

ADP/5/3/2/111-08

80334
809.

CR

UDR-TR-87-14

IMPROVEMENTS TO THE FASTEX FLUTTER ANALYSIS COMPUTER CODE

(NASA-CR-181072) IMPROVEMENTS TO THE FASTEX N87-24483
FLUTTER ANALYSIS COMPUTER CODE Final Report
(Dayton Univ.) 80 p Avail: NTIS HC
A05/MF A01 CSCL 01C Unclass
G3/08 0080334

Ronald F. Taylor
University of Dayton
Research Institute
Dayton, Ohio 45469

July 1987

Final Report
Grant NAG 2-377

Prepared For:

National Aeronautics and Space Administration
Ames Research Center
Hugh L. Dryden Flight Research Facility
Edwards, California



The University of Dayton

UDR-TR-87-14

IMPROVEMENTS TO THE FASTEX FLUTTER ANALYSIS COMPUTER CODE

Ronald F. Taylor
University of Dayton
Research Institute
Dayton, Ohio 45469

July 1987

Final Report
Grant NAG 2-377

Prepared For:

National Aeronautics and Space Administration
Ames Research Center
Hugh L. Dryden Flight Research Facility
Edwards, California

TABLE OF CONTENTS

SECTION		PAGE
1	INTRODUCTION	
1.1	Overview	1-1
1.2	Background	1-2
1.3	Objectives	1-3
1.4	Summary	1-3
2	EQUATIONS OF MOTION	
3	INTERPOLATION PROCEDURES	
3.1	One-Dimensional Spline	3-2
3.2	Two-Dimensional Spline	3-3
3.3	Polynomial Mode Representations	3-4
4	CONTROL SYSTEM CONSIDERATIONS	
5	SOFTWARE DEVELOPMENTS	
5.1	Overall Approach	5-1
5.2	Increased Problem Size Capability	5-14
5.3	Calculation of Aerodynamic Panel Data	5-18
5.4	Other Program Options	5-20
6	CONCLUSIONS	

REFERENCES

APPENDIX A:	INPUT INSTRUCTIONS FOR UDFASTEX
APPENDIX B:	PROCEDURE FILE DOCUMENTATION
APPENDIX C:	PLATE SPLINE THEORY

List of Figures

Figure		Page
5.1	Primary Code Modules in UDFASTEX	5-4
5.2	UDFASTEX Calling Sequence: MAIN	5-7
5.3	UDFASTEX Calling Sequence: POOL	5-8
5.4	UDFASTEX Calling Sequence: RODDEN	5-9
5.5	UDFASTEX Calling Sequence: SOLFLT	5-10
5.6	UDFASTEX Calling Sequence: MIDI, BIDI, MERGE	5-11
5.7	UDFASTEX Calling Sequence: FLOP	5-12
5.8	UDFASTEX Calling Sequence: EIGM	5-13

List of Tables

Table		Page
5.1	Option Selection Menu for Overall UDFASTEX System	5-2
5.2	Subroutines and Functions in UDFASTEX Code Modules	5-5
5.3	Key Words and Default Values for UDFASTEX Dimensions	5-15
5.4	Labeled Common Blocks in UDFASTEX	5-16
5.5	Include Files Used in UDFASTEX	5-17

FOREWORD

The work reported herein was performed in the Aerospace Mechanics Division of the University of Dayton Research Institute under Grant NAG 2-377 with the Hugh L. Dryden Flight Research Facility of the NASA Ames Research Center, Edwards, California. Dr. K. K. Gupta was the NASA project monitor.

The work described herein was conducted between December 1985 and June 1987, under the supervision of Mr. Dale H. Whitford, Supervisor, Aerospace Mechanics Division. Dr. Ronald F. Taylor was the Principal Investigator.

The author acknowledges the valuable discussions with Dr. Gupta and Mr. Len Voelker of NASA. Messrs. Ed Hahn and Roger Traux and Ali Ahmadi of Harvey Mudd College are also thanked for their assistance and suggestions. Mr. Tom Held of the University of Dayton Research Institute helped with programming and is also acknowledged for his contributions.

SECTION 1

INTRODUCTION

This final report documents the work completed by the University of Dayton Research Institute under a grant with the Dryden Research Facility of NASA Ames Research Center. The purpose of the grant was to further develop and evaluate the FASTEX flutter analysis computer code. Modifications and improvements to the code have been documented and are presented herein.

1.1 OVERVIEW

After a discussion of the general background of flutter analysis and the FASTEX code, Sections 2, 3, and 4 of this report document the relevant equations of motion, modal interpolation procedures, and control system considerations, respectively. This information is followed in Section 5 with a summary of the software developments. Additional information documenting input instructions, procedures, and details of the plate spline algorithm are found in the appendices.

The system of codes delivered to NASA is referred to as UDFASTEX. A magnetic tape, listings and sample runs are submitted under separated cover. These codes will allow NASA to make more efficient use of computer resources in the analysis of advanced aerospace vehicles. Key features of the codes are their improved modularity, more extensive internal documentation, and ease of interfacing new aerodynamic methods, and reduced user intervention in the interpolation of modal data. The UDFASTEX system operates on the VAX 11/700 series computers and makes use of user-friendly interactive menus written for operation under the VMS operating system.

1.2 BACKGROUND

In the development of advanced aerospace vehicles, it is essential to perform sufficient engineering calculations to assure that the vehicle will operate safely within its flight envelope. This is especially true for the major airframe components such as the wings, canards, and empennage. Calculations need to be performed to verify that these key airframe components have adequate strength and stiffness to withstand both the static and dynamic loadings.

Lifting surfaces must be designed to be aeroelastically stable and yet light enough to achieve overall performance requirements. It was with that objective in mind that the FASTOP computer code was originally developed in 1975 [1]. Both strength and flutter calculations are performed in FASTOP with the goal being to resize the structure so that near minimum weight can be achieved without violation of the constraints on stresses, deflections, and flutter speeds.

The FASTOP code was further improved since 1975 by the University of Dayton [2-4] and by Grumman Aerospace [5] by improving the optimization procedure and its structural analysis capability. However, by far the greatest use of the code has been to calculate flutter and divergence speeds. Its aeroelasticity capability has a number of state of the art features which make it a highly versatile flutter analysis tool.

In 1980 the Air Force Flight Dynamics Laboratory awarded a contract to the University of Dayton to streamline the FASTOP flutter analysis capability. The primary objective was to update the code so that it would operate only as an analysis tool without the optimization capability. The code was reorganized for faster execution times on the CYBER-175 system at Wright-Patterson Air Force Base. Additional features such as a more general modal interpolation procedure were also included. The resulting code, FASTEX [6], has since been extensively used in

both government and industry for the analysis of advanced flight vehicles such as the X-29A.

Although the code is now operational the VAX 11/780 minicomputer, further improvements were necessary to remove problem size limitations for operation on the CYBER. Modal interpolation between a structure grid and an aerodynamic grid have been improved to take advantage of the VAX virtual memory.

1.3 OBJECTIVES

The objectives of this grant have been to increase the problem size capacity of FASTEX, reduce run times by modification of the modal interpolation procedure, and to add new user features. Interfaces have been provided to aid in the inclusion of alternate aerodynamic and flutter eigenvalue calculations. Other user features have been added to increase the ease of program use.

1.4 SUMMARY

All modifications to the FASTEX code are operable on the VAX 11/700 series computers under the VMS operating system. The program can now be dimensioned to desired problem size and compiled. Aerodynamic and flutter solution methods are now operable as separate packages. Plots can be made of the flutter velocity, display and frequency data. A preliminary capability has also been developed to plot contours of unsteady pressure amplitude and phase.

SECTION 2

EQUATIONS OF MOTION

In this section the aeroelastic equations of motion are developed in modal coordinates. This is presented as basis for reference for the code modifications presented in Section 5.

Following the development of Meirovitch [7], we apply Hamilton's principle in the form

$$(2-1) \quad \int_{t_2}^{t_1} (\delta T + \delta W) dt = 0$$

where

$\delta W = -\delta V + \delta W_{NC}$ = virtual work

δT = variation of kinetic energy

δV = variation of potential energy

δW_{NC} = virtual work due to nonconservative forces.

The modal assumptions

$$(2-2) \quad w = \sum_{i=1}^N \phi_i(x,y) q_i(t)$$

$$(2-3) \quad p = \sum_{i=1}^N p_i(x,y) q_i(t)$$

are made where

$w = w(x,y,t)$ = lateral deflection at point x,y at time t

p = net upward air load at point x,y at time t

ϕ_i = assumed deflection mode

p_i = pressure contribution due to ϕ_i mode

q_i = generalized coordinate.

This leads to the kinetic energy expression

$$(2-4) \quad T = \frac{1}{2} \{\dot{q}\}^T [M] \{\dot{q}\}$$

where the elements of the $[M]$ matrix are

$$(2-5) \quad M_{ij} = \iint_S m(x,y) \phi_i(x,y) \phi_j(x,y) dS$$

$i=1, \dots, N; j=1, \dots, N$

with

m = mass per unit area of the lifting surface

S = surface area.

Also, the potential or elastic strain energy takes the form

$$(2-6) \quad V = \frac{1}{2} \{q\}^T [K] \{q\}$$

where the elements of the $[K]$ matrix are

$$(2-7) \quad K_{ij} = \iint_S k(x,y) \phi_i(x,y) \phi_j(x,y) dS$$

and

$k(x,y)$ = stiffness influence function.

The final term in the formulation requires the variation of the nonconservative work term which has the form

$$(2-8) \quad \delta W_{NC} = \{\delta q\}^T [P] \{q\}$$

where the terms in the pressure matrix are

$$(2-9) \quad P_{ij} = \iint_S \phi_i(x,y) p_j(x,y) dS$$

Taking the variation of the kinetic and strain energies and using Equations 2-8 and 2-9 in Equation 2-1 leads to the equation of motion

$$(2-10) \quad [M]\{\ddot{q}\} + [K]\{q\} - [P]\{q\} = \{0\}$$

Introduction of the assumption of harmonic motion and hysteretic damping gives

$$(2-11) \quad -\omega^2[M]\{\bar{q}\} + (1+ig)[K]\{\bar{q}\} = [P]\{\bar{q}\}$$

where the [P] matrix is determined using one of the conventional unsteady aerodynamic programs such as doublet lattice. The [P] matrix is expressed as

$$(2-12) \quad [P] = \frac{1}{2}\rho V^2[\bar{Q}]$$

and substitution into (2-11) yields

$$(2-13) \quad \lambda[K]\{\bar{q}\} = ([M] + [Q*])\{\bar{q}\}$$

where

$$\lambda = (1 + ig)/\omega^2$$

$$[Q*] = c[\bar{Q}]$$

$$c = \frac{1}{2} \rho V^2/\omega^2$$

Note that the reduced frequency is

$$k = \omega b/V$$

so [Q*] becomes

$$(2-14) \quad [Q*] = (c/1/2\rho V^2)[P]$$

Since $[\bar{Q}]$ is the generalized force per unit dynamic pressure

$$(2-15) \quad [\bar{Q}] = d [Q^*]$$

where $d = 1/c = 2k^2/\rho b^2$. The aerodynamic forces output by the UDFASTEX routine called MATDAT.

The flutter eigenvalue problem solved is

$$(2-16) \quad [A]\{\bar{q}\} = \lambda[B]\{\bar{q}\}$$

where

$$[A] = [M] + [Q^*]$$

and

$$[B] = [K]$$

Since $[Q^*]$ is a function of reduced frequency, k , and Mach number, M , the $[A]$ matrix is also

$$[A] = [A(k,M)].$$

SECTION 3

INTERPOLATION PROCEDURES

Several situations arise in a flutter analysis where it is necessary to perform interpolations on given or computed data. These interpolations generally involve data of either one or two variables. The discussion in this section is directed toward the following three common interpolation problems:

- Evaluation of aerodynamic influence coefficient matrices at intermediate values of reduced frequencies for a fixed Mach number (i.e., one-dimensional interpolation).
- Evaluation of aerodynamic influence coefficient matrices at intermediate pairs of reduced frequency and Mach number.
- Calculation of modal deflections and slopes at the aerodynamic grid points in terms of data based on the structural model.

The first interpolation problem alone is an essential feature of the p-k method. Interpolation of modal data is required in nearly all problems with the exception of piston theory calculations. In the piston theory case, aero and structural grid points can be easily made identical. The use of doublet-lattice or Mach box methods requires a specialized grid which rarely is the same as the structures grid.

In the following subsections, spline and polynomial interpolation procedures are discussed. It is generally considered that splines yield less oscillatory interpolated values than polynomial approximations based on least-squared fits. Polynomial approximations are discussed since many of the air loads programs (see References 8 and 9, for example) require as input the coefficients to polynomials which approximate the mode shapes.

3.1 ONE-DIMENSIONAL SPLINE

A one-dimensional cubic spline is a piecewise polynomial defined on an interval (x_1, x_N) . Functional values are assumed to be known at a given set of points x_1, x_2, \dots, x_N which are called knots. The polynomial is so constructed that its functional values equal the given values at the knots and furthermore its first and second derivatives are continuous at these points. The work by deBoor [10] provides much useful information regarding both the theoretical and practical aspects of spline interpolation.

To evaluate aerodynamic matrices at intermediate values of reduced frequencies, use can be easily made of IMSL routines [11]. For a given real or imaginary term in an aerodynamic matrix, its value is known for a given number of reduced frequencies

k_1, k_2, \dots, k_n . It is desired to determine the value of this term for some value of reduced frequency, \bar{k} , which lies between k_1 and k_n . This is accomplished by two IMSL calls. The first is to the routine ICSCCU which obtains the values of the coefficients to the spline polynomial. The second call is to ICSEVU which uses the coefficients to compute the given term at the selected value of \bar{k} . Loops must be set up to evaluate all terms, both real and imaginary, in the aerodynamic matrix. Numerical experimentation using the IMSL routines, together with Reference 8 doublet lattice program, has shown that quite accurate results can be obtained for a given set of widely spaced k values.

The above procedure can also be used to evaluate aerodynamic matrices when both reduced frequency and Mach number are treated as variables. Assume that k_1, \dots, k_n are the given reduced frequencies and that M_1, \dots, M_m are the Mach numbers. Thus the aerodynamics are known for all pairs (k_i, M_j) for $i=1, \dots, n$ and $j=1, \dots, m$. To find the aerodynamics at any (\bar{k}, \bar{M}) pair which lies within the (k_i, M_j) grid the following procedure is suggested for any given real or imaginary term in the aerodynamic matrix:

1. Set $M=M_1$ and find the value of the term at $k=\bar{k}$ using the above IMSL routines. Save this value.
2. Repeat Step 1 for $M=M_2, \dots, M_m$.
3. Using the saved values from Steps 1 and 2, find the value of the term at $M=\bar{M}$ using the IMSL routines, treating M as the variable instead of k .
4. Repeat Steps 1, 2, and 3 for all terms in the matrix, both real and imaginary parts.

By the judicious use of interpolation on the aerodynamic matrices, experience has shown that computing costs for a flutter analysis can be reduced by as much as a factor of 10. This is especially true for complex configurations when more than 10 modes are used in the flutter analysis.

3.2 TWO-DIMENSIONAL SPLINE

The IMSL library also contains a routine, IBCCU, for interpolating data of two variables. Based on experience with interpolation of modal data given an (x,y) grid, special purpose procedures should be used rather than routines like IBCCU. In the following paragraphs a special purpose "plate spline" is discussed and accuracy improvement features are suggested.

The interpolation problem is thus stated:

Given

$$(x_j, y_j, w_j) \quad , \quad j=1, \dots, N$$

and

$$(\bar{x}_j, \bar{y}_j) \quad , \quad j=1, \dots, M$$

fit a mathematical function $W=W(x,y)$ through the points W_j and compute

$$W(\bar{x}_j, \bar{y}_j) = W_j$$

$$\frac{\partial}{\partial x} W(\bar{x}_j, \bar{y}_j) = W_{xj} \text{ (optional)}$$

$$\frac{\partial}{\partial y} W(\bar{x}_j, \bar{y}_j) = W_{yj} \text{ (optional)}$$

The data (x_j, y_j, W_j) can be coordinates and either given modal deflections or given modal slopes in the streamwise direction. It is recommended that if modal slopes are available in the structures grid, this should be used rather than the optional W_{xj} calculations. Appendix C outlines the basic bivariate spline procedure based on the work of Harder and Desmarais [12]. Additional features that are not in the IMSL bivariate method are

1. Translation of given data to a centroidal origin,
2. Rotation of data to principal axes,
3. Scaling of x-y data by the reciprocal of their maximum values,
4. Linpack [13] is used to solve the equation.

Use of this procedure gives quite accurate normalwash information for input to the aerodynamic routines. Whenever possible, it is recommended that structural grid data be used which extends as nearly as possible to the perimeter of the aerodynamic planform. The plate spline can extrapolate beyond the structures grid; however, the streamwise slopes may not be accurately represented.

3.3 POLYNOMIAL MODE REPRESENTATIONS

Programs such as the doublet-lattice [8] require that vibration mode amplitudes be input in the following nondimensional form

$$(3-1) \quad \frac{w^{(k)}(x,y)}{b} = \sum_{i=1}^n \sum_{j=1}^m a_{ij}^{(k)} \left(\frac{x}{b}\right)^{i-1} \left(\frac{y}{b}\right)^{j-1}$$

where the k superscript denotes the mode number, b is the reference length used in the reduced frequency definition, $a_{ij}^{(k)}$ are coefficients to be determined, and n and m are integers which determine the degree of the polynomial in x and y , respectively. For a given set of r discrete deflections in the k^{th} mode $(\phi_1^{(k)}, \dots, \phi_1^{(k)})$ at the known points $((x_1, y_1), \dots, (x_r, y_r))$, Equation 3-1 can be put in the matrix form

$$(3-2) \quad \{\phi^{(k)}\} = [B] \{a^{(k)}\}$$

where $[B]$ is a rectangular matrix of size $r \times p$ and $p=nm$. The coefficients of Equation 3-1 have been arranged in a column vector and the $[B]$ matrix contains terms involving powers and products of the known point coordinates.

In the usual case when $r > p$, Equation 3-2 is an over-determined system. In this case the standard least-squares error approach can be used to determine the coefficients. The IMSL routine for this problem is LLSQF.

When the least-squares polynomial approximation is used, great care should be taken to use as low an order polynomial as possible to represent the modes. The possible oscillatory behavior of high order polynomials can cause highly inaccurate values of streamwise slope which will lead to unrealistic pressure distributions.

SECTION 4

CONTROL SYSTEM CONSIDERATIONS

The use of active control systems to increase the operational range of flight vehicles which are subject to aeroelastic instabilities has been an emerging technology since the early 1970's. The signals from sensors can be used to command the movement of aerodynamic controls to prevent flutter [14]. The use of velocity-root locus plots in conjunction with the p-k method is a feasible approach. Both rigid body and flexible mode variations need to be included in the active control method. The work of Nissim [15-16] and Rudisill [17] is important from the standpoint of setting up an optimal control law which relates the system response to the modal deflections.

The approach used in NASTRAN is one approach to general control system problems. The viewpoint is that the control system is accessory to the structure. The vector of structural degrees of freedom is augmented by the addition of extra variables which are particular to the control systems (e.g., voltages, position variables, etc.). It is assumed that these control system variables, denoted by u_e , are related to N structural degrees of freedom or control variables, u_i , by the following expression

$$(4-1) \quad u_e = \frac{1}{b_0 + b_1 p + b_2 p^2} \sum_{i=1}^N (a_0^{(i)} + a_1^{(i)} p^2) u_i$$

where the a's and b's are user specified constants and p is the first order time derivative operator.

Much of the work on active flutter control has utilized the motion of flaps, tabs, and other leading and trailing edge control devices. It is feasible also to utilize aerodynamic interference effects due to adjacent surfaces. By input of appropriate torsional root motion to the forward surface, for

example, a flutter condition on the aft surface may be controlled.

SECTION 5

SOFTWARE DEVELOPMENT

This section describes the software modifications that were performed on a NASA-developed version of FASTEX. The NASA code used for this work is a modified form of the Reference 6 codes which were developed for a CYBER computer system. The primary NASA modifications to FASTEX dealt with file structures, program interfaces, and related coding changes required for operation under the VMS operating system on VAX 11/700 series computers. The system of codes described herein are referred to collectively as UDFASTEX.

5.1 OVERALL APPROACH

The overall approach taken to improving the FASTEX code was to reorganize the code into major program modules, provide for an improved modal interpolation methodology, and develop a convenient user interface. A single procedure manages the files and provides a menu-based interface for activating all the program options.

Table 5.1 shows the UDFASTEX main menu. The various options are described in the subsections below. Appendix B documents the procedure file which generates this menu. When options are selected from the menu, further information is given concerning files which are needed to execute the various options.

Figure 5.1 shows the primary program modules:

UDFASTEX1	Collection of frequently called subroutines and the main calling program. This module is used when Option 1 is selected from the menu.
UDFASTEX2	Same as above, except that it is designed for use with Option 2.
POOL_ET_AL	Routines for basic flutter input.

Table 5.1

Option Selection Menu for Overall UDFASTEX System

UDFASTEX MAIN MENU

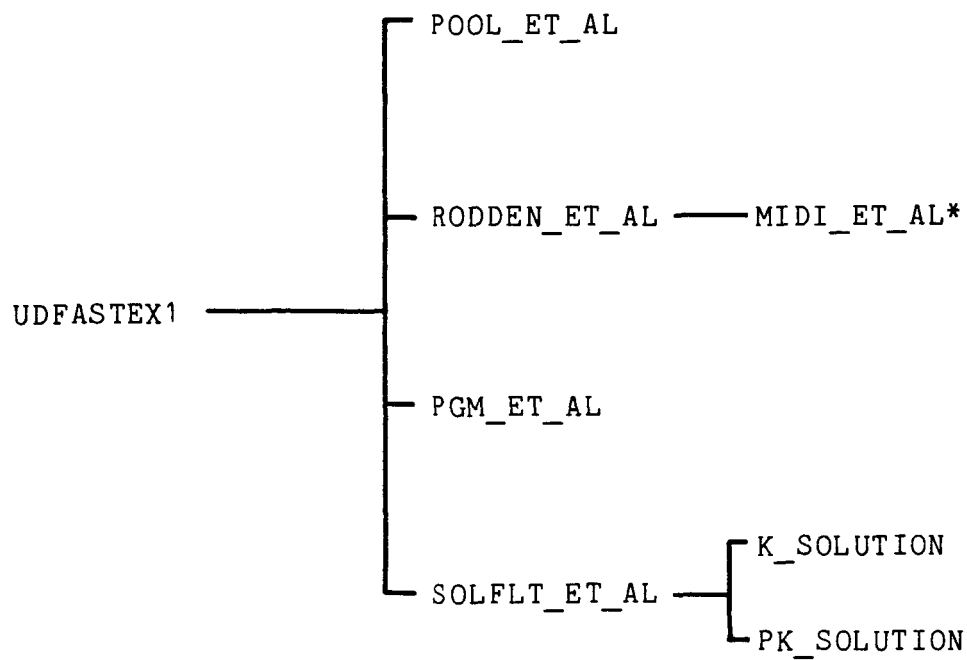
OPTION	DESCRIPTION
1	Run UDFASTEX with internal mode interpolation
2	Run UDFASTEX with external mode interpolation
3	Compute aero box data or run plate spline
4	Run an alternate aero or eigen solution
5	Plot flutter V-g or V-f data
6	Change the problem size capacity of UDFASTEX
7	Compile and link an new version of UDFASTEX for OPTION 1 use
8	Compile and link an new version of UDFASTEX for OPTION 2 use
9	Create a print file of selected UDFASTEX output
99	Quit

ENTER OPTION

RODDEN_ET_AL	Subsonic aerodynamic routines.
MIDI_ET_AL	Reference 6 FASTEX modal interpolation routines.
NEW_MIDI_ET_AL	Routine which reads the downwash output file from the spline MODE_INTERPOLATION program.
PGM_ET_AL	Main calling routine for a supersonic aerodynamic code.
SOLFLT_ET_AL	Main calling routine for flutter eigensolution.
K_SOLUTION	k-method flutter solution routines.
PK_SOLUTION	p-k method flutter solution routines.

Table 5.2 lists the subroutines and functions included in these modules. The NEW_MID_ET_AL module is not shown since it calls no subroutines. The program modules are files which are compiled and linked to form the UDFASTEX1 and UDFASTEX2 executable files.

The calling sequence of all the subroutines in the UDFASTEX codes shown in Table 5.2 is given in Figures 5.2 through 5.8.



*NEW_MIDI_ET_AL for UDFASTEX2 System.

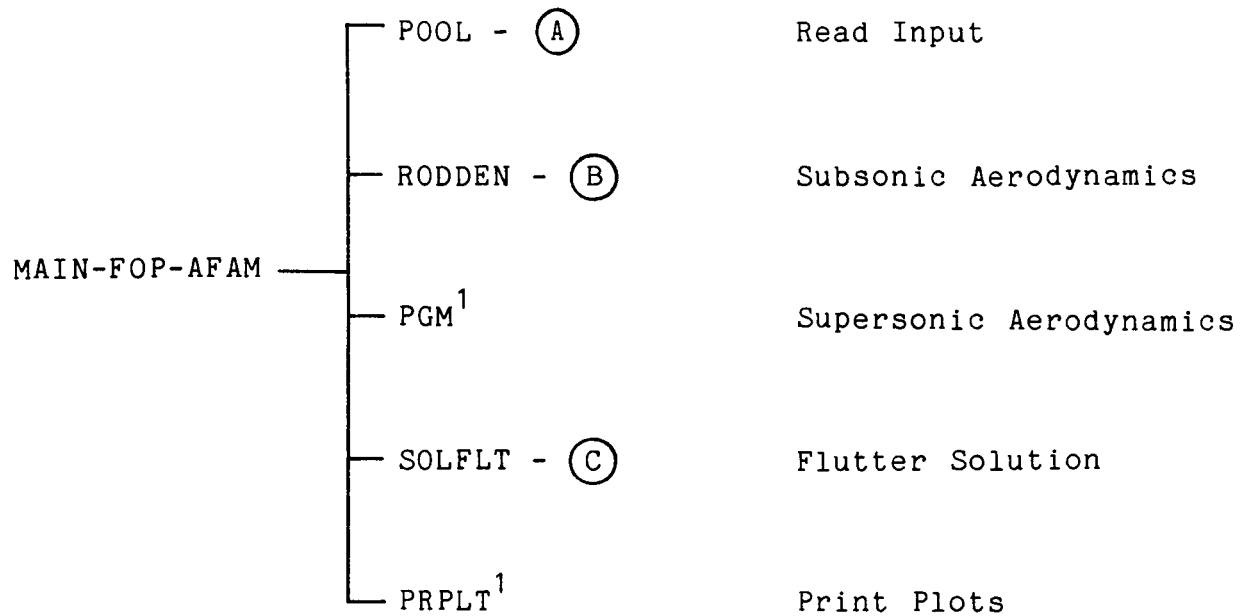
Figure 5.1. Primary Code Modules in UDFASTEX.

TABLE 5.2
SUBROUTINES AND FUNCTIONS IN UDFASTEX CODE MODULES

Module	Subroutines	Functions
UDFASTEX1 and UDFASTEX2	FOP, AFAM, AORDER CLUSAL, CLUSTL, COMSCA1, CONV, DVALUE, HELGX, ORIENT, QFLIN, RNRW	ANDOR, CDABSF, CDAT, CDET, COMSCA, DCMPLF, DSCAPR, DSQRTF, F, RDM, ROUND
POOL_ET_AL	POOL, FLINFO, DENSITY	-
RODDEN_ET_AL	RODDEN, PART1, BIDI, MERGE, GLOBAL, GENQ, PRT2, QUAS, FUTSOL, GENF, QINTP1, ATAN3, BEIN, AUGW, INCRO, SNPDI, SPLIT3, TRIDI, TKER, KERNEL, IDF1, IDF2	-
PGM_ET_AL	PGM	-
SOLFLT_ET_AL	SOLFLT, PRPLT, PICTUR, SCLMAX, SCLINC	-

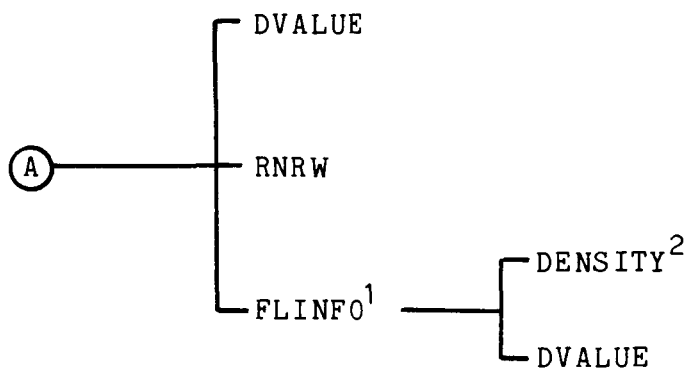
TABLE 5.2 (continued)

Module	Subroutines	Functions
MIDI_ET_AL	MIDI, MODAL, FORM, HELGA, HELP, MOVIS	-
K_SOLUTION	EIGM, SREVNC, FLISL, BUCK, GGCHK, TRFR, FLASH, VALCOM, VALROM, COMVEC, CQR, CLR, CLINEQ, MATDAT	-
PK_SOLUTION	FLOP, RTIN, ZANLYN, FRORD, VECP, ADIV, GCVEC, GRVEC, ASSESS, UERTST, GENEIG, JORCOM	-



¹Inactive in Current UDFASTEX.

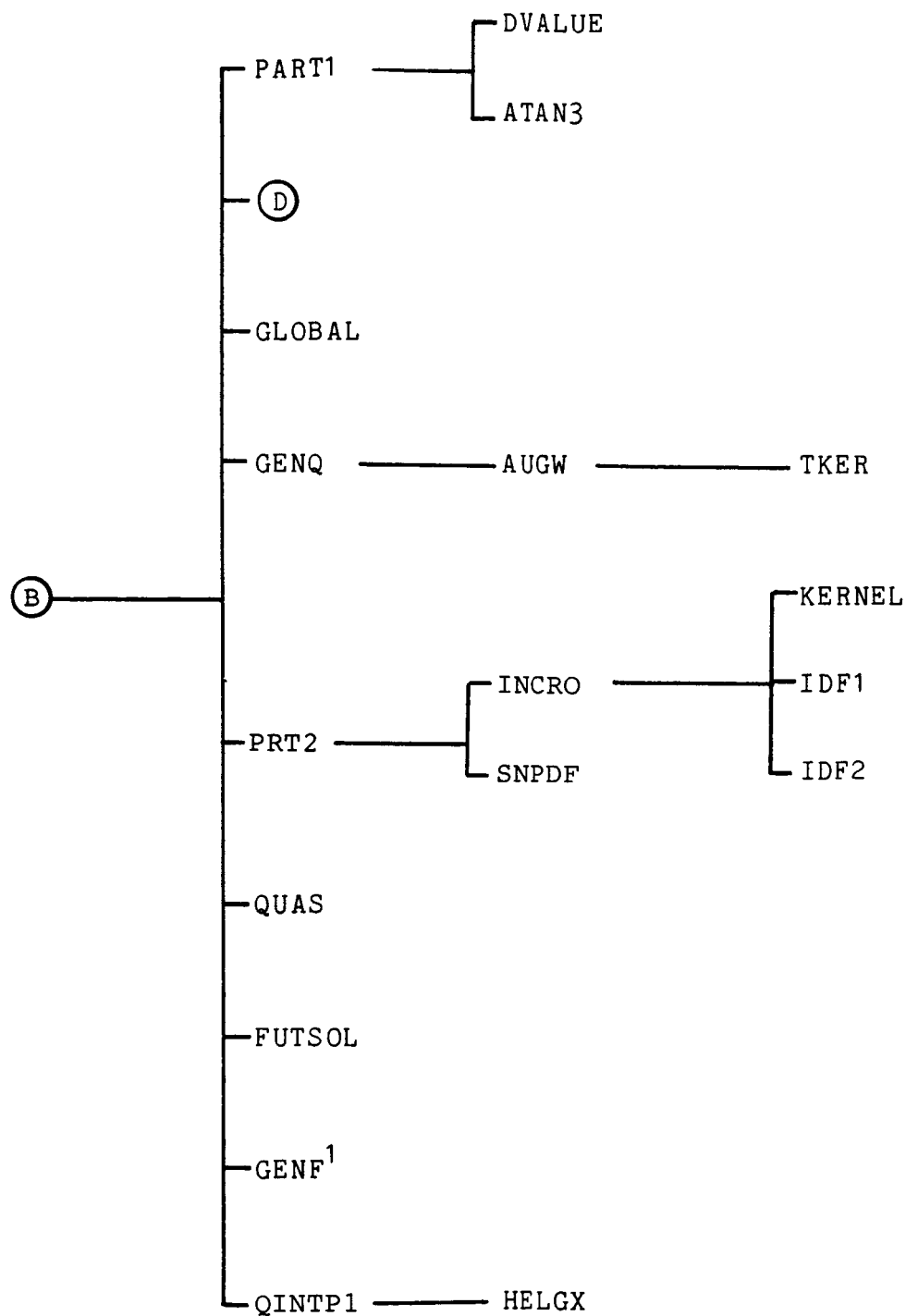
Figure 5.2. UDFASTEX Calling Sequence: MAIN.



¹ All of the basic flutter data (excluding aerodynamic and modal) are input in FLINFO.

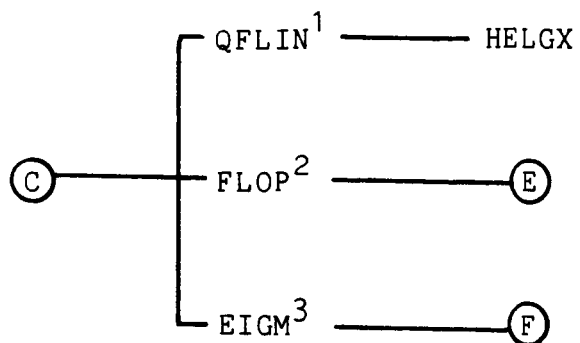
² New routine to compute air density for given altitude.

Figure 5.3. UDFASTEX Calling Sequence: POOL.



¹Generalized aerodynamic forces are printed by GENF.

Figure 5.4. UDFASTEX Calling Sequence: RODDEN.

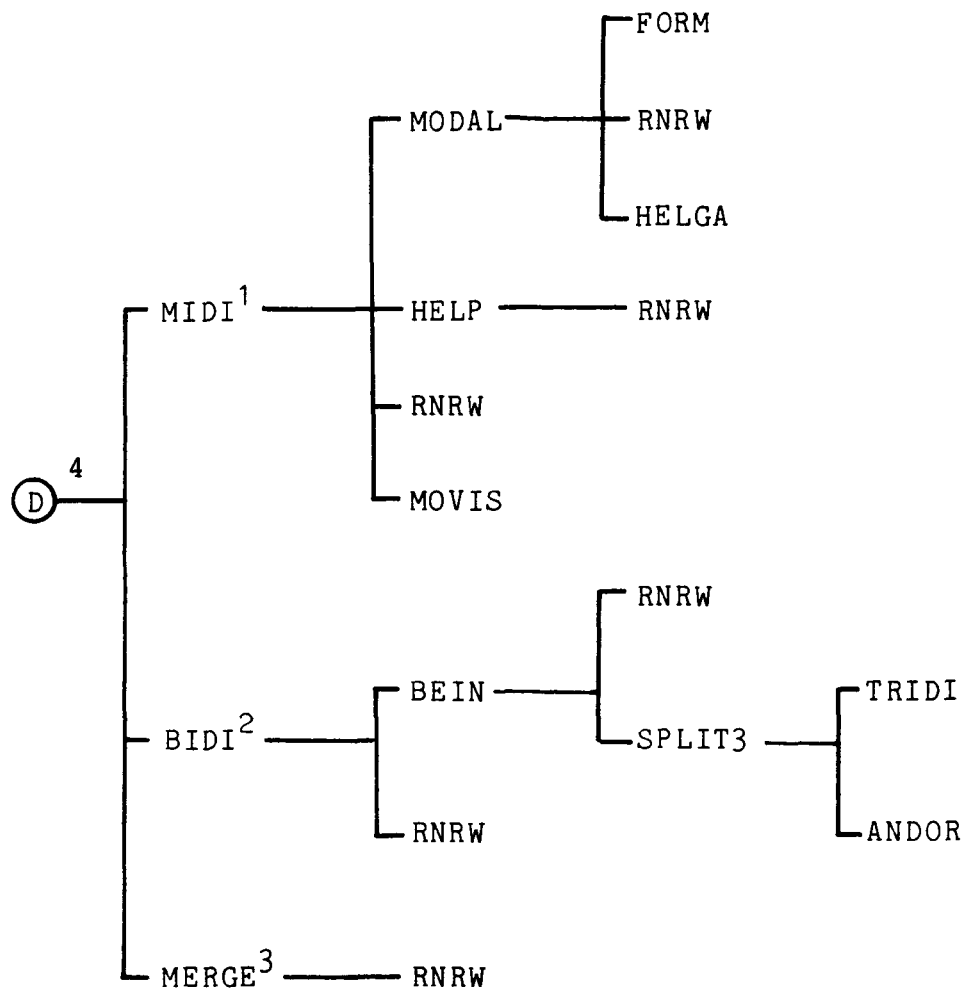


¹Generalized aerodynamic forces are printed by QFLIN

²p-k flutter calculations

³k flutter calculations

Figure 5.5. UDFASTEX Calling Sequence: SOLFLT.



¹ Surface modes interpolated

² Body modes interpolated

³ Downwash file assembled for surfaces and bodies

⁴ Called only for UDFASTEX1. Separate MODE_INTERPOLATION program used for UDFASTEX2.

Figure 5.6. UDFASTEX Calling Sequence: MIDI, BIDI, MERGE.

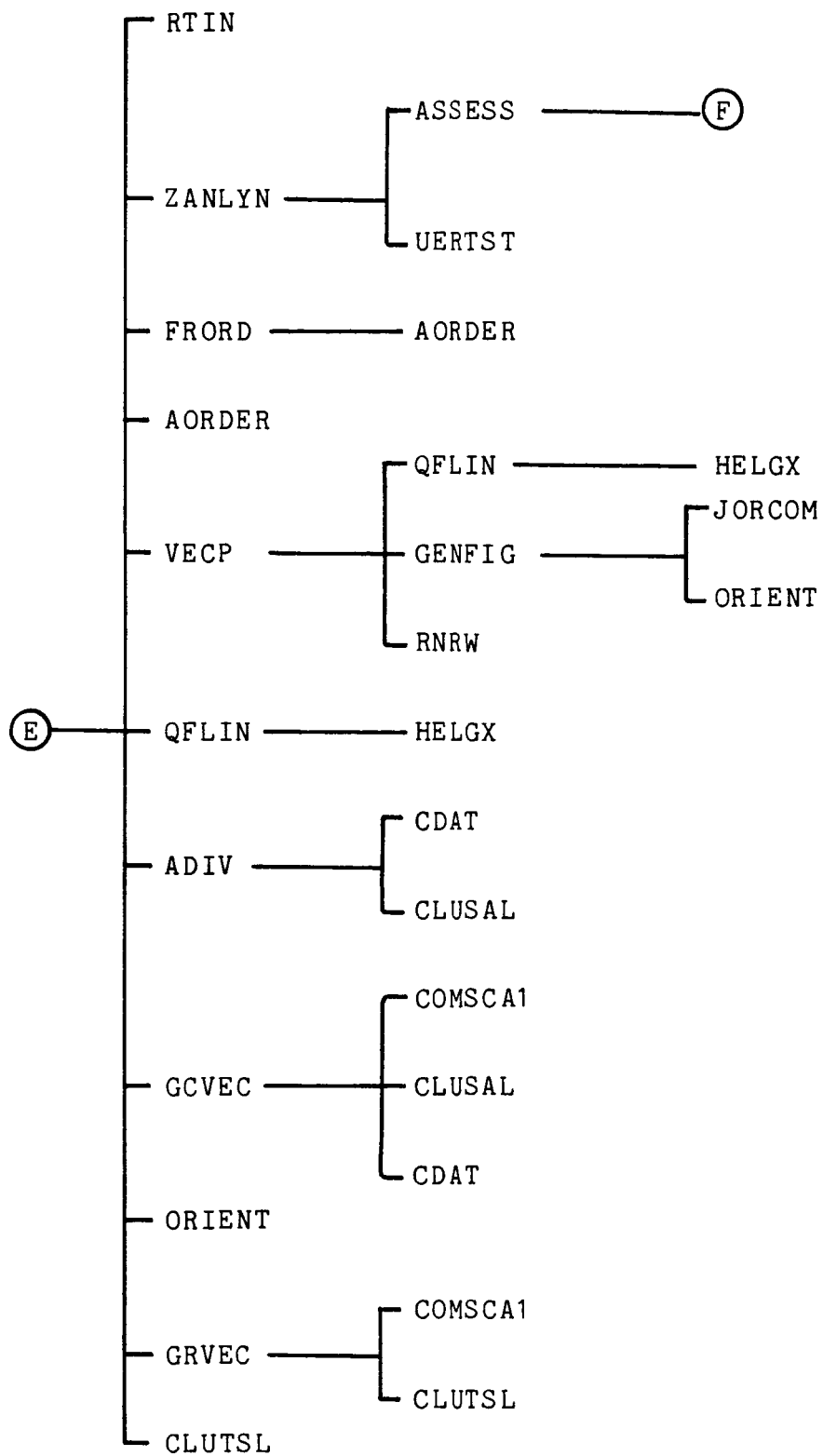
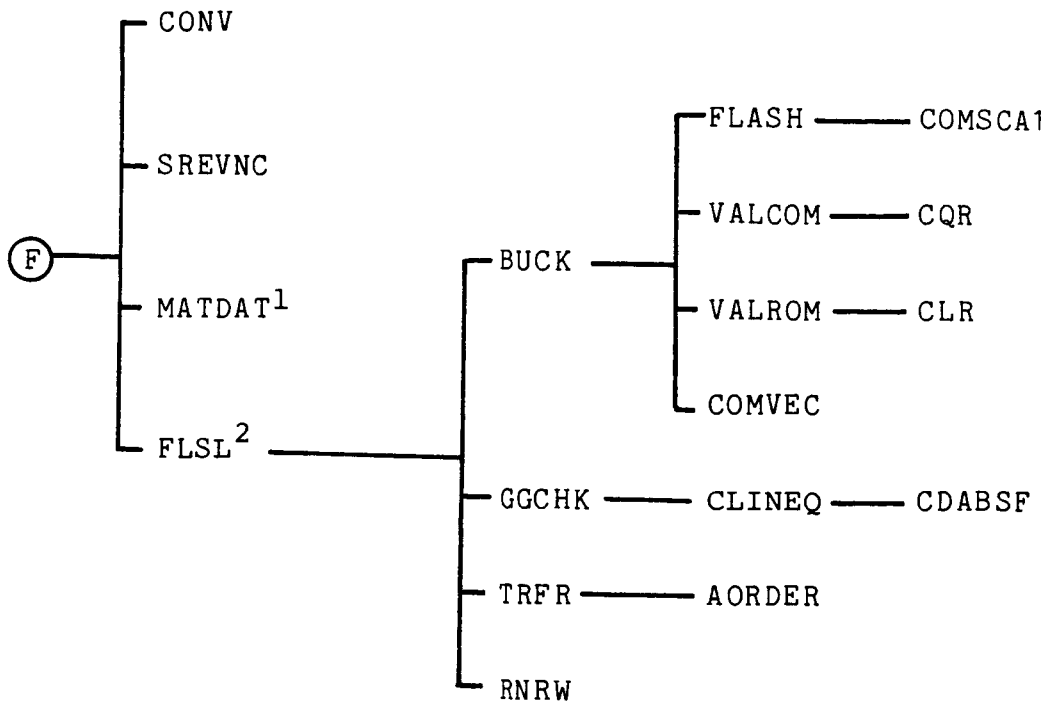


Figure 5.7. UDFASTEX Calling Sequence: FLOP.



¹Creates output file with all matrices required to perform an external flutter eigensolution.

²Prints k-flutter solution results.

Figure 5.8. UDFASTEX Calling Sequence: EIGM.

5.2 INCREASED PROBLEM SIZE CAPACITY

When Option 6 is selected from the Main Menu, a Fortran code, SIZE, is executed which prepares a batch edit file for modifying the UDFASTEX "Include Files." These files are designated by the file type, \$.NS, and they contain Key Words which are replaced with numerical values which determine the dimensioning of the program. When the batch editing job is completed, the Include Files have the file type, .INS.

Table 5.3 lists the Key Words, their description, and default values. The SIZE program is interactive and displays the description and default values shown in the table. The last column of Table 5.3 refers to the data item and page number in the original FASTOP manual [1] which is still the basis for most of the UDFASTEX input.

The UDFASTEX labeled common blocks are shown in Table 5.4. A majority of these contain the Key Word parameters which are replaced with numerical values. The Include Files contain dimensions statements and other variables which are set by the SIZE code. Table 5.5 is a listing of all the Include File names.

Table 5.3

Key Words and Default Values for UDFASTEX Dimensions

<u>Key Word</u>	<u>Description</u>	<u>Default</u>	<u>Item/Page</u>
\$MOD	Modes, LC(2)	20	4/218
\$SUF	Surfaces, LC(3)	30	4/218
\$REV	Reduced Velocities, LC(4)	30	4/218
\$DEN	Air Densities, LC(5)	10	4/219
\$MEC	Mode Elimination Cycles, LC(25)	25	4/221
\$SVC	Stiffness Variation Cycles, LC(26)	20	4/221
\$PBA	Max p-k Velocities	20	19/227
\$PAN	Total Panels, NP	50	3R/239
\$BOD	Total Bodies, NB	20	3R/240
\$SOP	Problem Size, NCORE	400	3R/240
\$CON	Total Control Surfs	5	16R/246
\$MOL	Modal Lines, NGP	12	18R/247
\$PBA	Body Points, NGP	20	31R/252

Table 5.4

Labeled Common Blocks in UDFASTEX

BAL	COLMEM	DETAIL	LABELS
BANDW	COMA	DLM	MATRIX
BODY	COMRWP	DSIO1	MESAG
CALCP	CONSTS	DSIO2	MODD
CBYTES	CORE	DSRN	MODE
CDATER	CPLOTS	ELMNT	MODV
CEELO2	CPLOTF	FACE	MODV
CFILES	CSETUP	FILE	NTPS
CFMTA	CSHIFT	FITR	P1GW
CFMTBO	CTABLE	FLEXT	PLACES
CFMTB	CTAPES	FLUT	PLAYFF
CFMTCO	CTFH	FLUTAN	PLUG
CHEAD	CTITLE	FLUTB	PRPL
CHSP	CTMH	FLUTC	REPORT
CIDIV	CTSH	FLUTQ	SIZES
CLIST	CTSHF	FLUTV	VARBLS
CLUEF	CTSHF0	FSIO1	WAYS
CLUEM	CTSHV	FSIO2	XYZ
CLUEV	CVIBRA	JUNK	YZY
CLUFO	DCOM1	KLUES	
CNTRL	DCOM2	KLUFF	
COLDES	DCOM3	KMP	

Table 5.5

Include Files Used in UDFASTEX

ADIV	AFAM	ANDOR	AORDER
ASSESS	ATAN3	AUGW	BEIN
BIDI	BUCK	CDABSF	CDAT
CDET	CLINEQ	CLR	CLUSAL
CLUTSL	COMSCA	COMSCA1	COMVEC
CONV	CQR	DCMPLF	DSCAPR
DSQRTF	DVALUE	EIGM	F
FLASH	FLINFO	FLOP	FLSL
FOP	FORM	FRORD	FUTSOL
GCVEC	GENEIG	GENF	GENQ
GGCHK	GLOBAL	GRVEC	HELGA
HELGX	HELP	IDF1	IDF2
INCRO	JORCOM	KERNEL	MATDAT
MERGE	MIDI	MODAL	MOVIS
NEW MIDI	ORIENT	PART1	PGM
PICTUR	POOL	PRPLT	PRT2
QFLIN	QINTP1	QUAS	RDM
RNRW	RODDEN	ROUND	RTIN
SCLINC	SCLMAX	SNPDF	SOLFLT
SPLIT3	SRENV	TKER	TRFR
TRIDI	UDFASTEX1	UDFASTEX2	UERTST
VALCOM	VALROM	VECP	ZANLYN

5.3 CALCULATION OF AERODYNAMIC PANEL DATA

Option 3 on the Main Menu has two functions. The first is the calculation of aerodynamic panel and body data. The second is the execution of the plate spline modal interpolation. This subsection deals with the first Option 3 function which must be completed before the interpolation program is executed. The next subsection deals with the modal interpolation part of Option 6.

The calculation of the aerodynamic panel and body data is accomplished by the program AERO_BOX. The panel data consists of the coordinates of the sending and receiving points plus the length and width of each panel. See Reference 8 for a discussion of the theory behind the aerodynamic paneling process. Reference 18 contains the documentation for the doublet lattice computer code, H7WC. This code is essentially the same as the one currently in UDFASTEX and the previous versions of FASTEX and FASTOP. Figure 5.9 shows a sketch from Reference 18 and also the variable names for the key output of AERO_BOX.

The input to AERO_BOX is the usual UDFASTEX_INPUT.DAT file. The output files created are

```
AERO_BOX_OUTPUT.DAT
AERO_BOX_XYZ_SENDING.DAT
AERO_BOX_XYZ_RECEIVING.DAT
AERO_BOX_LENGTH_WIDTH.DAT
AERO_BOX_BODY.DAT
BEIN_INPUT.DAT
```

These contents of these files are described in Table 5.6.

The AERO_BOX program contains the UDFASTEX code as the main program. Calls to the flutter solution routine, SOLFLT, have been removed. The only code from RODDEN_ET_AL that is used is PART1 and its associated subroutines. Subroutines RODDEN, PART1,

The AERO_BOX program contains the UDFASTEX code as the main program. Calls to the flutter solution routine, SOLFLT, have been removed. The only code from RODDEN_ET_AL that is used is PART1 and its associated subroutines. Subroutines RODDEN, PART1, and POOL are renamed AERO_BOX_RODDEN, AERO_BOX_PART1, and AERO_BOX_POOL respectively. The POOL_ET_AL routines have been modified so that the downwash file FOR049.DAT is not created.

The MODE_INTERPOLATION program theory is documented in Appendix C. The main routine contains an overall loop on surface data and an inner loop on modes. The surfaces are interpolated first, followed by a call to BIDI_MAIN which performs the body mode interpolation.

The interpolation requires as input all of the above AERO_BOX files except AERO_BOX_OUTPUT.DAT. The user must also prepare files as follows:

MODE_POINTS_XYZ.DAT
MODE_SHAPES_DEF.DAT
BODY_MODES.DAT

These files contain the given modal data and mode shape deflections for the surfaces and fuselage. The formats are included in the internal program documentation of MODE_INTERPOLATION. Appropriate modifications can easily be made to MODE_INTERPOLATION to accommodate input generated by a finite element code. Some sorting of the data will be required in order to eliminate all but the deflection components normal to the surface.

Output files from MODE_INERPOLATION are FOR041,42,..., FOR0XX, FOR027, FOR034 where XX = 40 + number of surfaces. A TOTALS.DAT file is also created which contains problem size data.

The MODE_INTERPOLATION program is intended to be run before execution of UDFASTEX2. All files are managed by the UD.COM procedure. The user is presented menus and directories indicating which files are needed for MODE_INTERPOLATION and UDFASTEX2.

5.4 OTHER PROGRAM OPTIONS

Other program options provide for generation of flutter plots (Option 5), independent flutter eigensolution (Option 4), and compiling and linking of the various UDFASTEX versions of the code (Options 7 and 8). An Option 9 has also been included to selectively print portions of the UDFASTEX output file. This has not been fully implemented; however, as user requirements are defined for particular problem output requirements the option can be made fully active by studying the comments in the UD.COM file.

The flutter solution requires the output file of the MATDAT subroutine and is fully compatible with the UDFASTEX file structures. The solution method is based on the LZ algorithm and solves the generalized complex eigenvalue problem without matrix inversion.

SECTION 6

CONCLUSIONS

A modified version of FASTEX has been developed and documented. The key developments are dynamic dimensioning of arrays and an improved modal interpolation methods for the lifting surfaces.

Additional work is need to fully checkout the modal interpolation for the fuselage. Several users have expressed concern about the accuracy of the interpolated results. In further checking of the problem, it appears that the original version of FASTOP have have not been accurately coded. Additional work is needed to verify this. As it currently exists, the UDFASTEX code is fully operational for models with multiple lifting surfaces. It can also be used for unsymmetrical configurations by following the input requirements in Appendix A herein.

Due to the complexity of the code and its file structures, further development of FASTEX is questionable. It is important to recognize that UDFASTEX is a fully developed and operational code except as noted above. In view of new demands imposed by upcoming aerospace system, it appears highly desirable to initiate planning of a new generation of flutter analysis tools which will take advantage of new and efficient computer hardware.

New code developments should pay particular attention to data structures and modern structured programming. Standard mathematical software has developed to a very high degree since the original FASTOP code development was planned in the early 1970's. Utilization of this software should be planned in the next generation codes.

Advances in computational fluid dynamics should also be reviewed. Panel methods are highly efficient; however, flutter

behavior in the transonic regime cannot be accurately predicted with current flutter analysis tools. New CFD approaches as well as improved panel methods need to be developed and added to a structured flutter analysis code.

Alternatives to conventional eigensolutions should be developed. Nonlinear effects can be handled with time integration of the basic equations. This can also be used in the next generation of codes in which active controls are included.

REFERENCES

1. Wilkinson, K., et al., "An Automated Procedure for Flutter and Strength Analysis and Optimization of Aerospace Vehicles," AFFDL-TR-75-137, Vols. I and II, December 1975.
2. Brockman, R. A., "Modification of the Structural Optimization Code in FASTOP," UDR-TR-77-51, University of Dayton Research Institute, Dayton, Ohio, September 1977.
3. Brockman, R. A., "Automated Conversion of NASTRAN Finite Element Data for the FASTOP Program," UDRI-TM-77-02, University of Dayton Research Institute, Dayton, Ohio, March 1977.
4. Brockman, R. A., "Modification of the Flutter Optimization Code in FASTOP," UDR-TR-77-60, University of Dayton Research Institute, Dayton, Ohio, October 1977.
5. Markowitz, J., and Isakson, G., "FASTOP-3: A Strength, Deflection, and Flutter Optimization Program for Metallic and Composite Structure," AFWAL-TR-78-50, Vols. I and II, May 1978.
6. Taylor, R. F., Miller, K. L., and Brockman, R. A., "A Procedure for Flutter Analysis of FASTOP-3 Compatible Mathematical Models," AFWAL-TR-81-3063, Vol. I and II, June 1981.
7. Meirovitch, L., Analytical Methods in Vibration. Macmillan Company, New York, 1967.
8. Albano, E., and Rodden, W. P., "A Double-Lattice Method for Calculating Lift Distributions on Oscillating Surfaces in Subsonic Flows," AIAA Journal, Vol. 7, No. 2, February 1969, pp. 279-285; errata, AIAA Journal, Vol. 7, No. 11, November 1969, p. 2192.
9. Donato, V. W., and Huhn, C. R., "Supersonic Unsteady Aerodynamics for Wings with Trailing Edge Control Surfaces and Folded Tips," AFFDL-TR-68-30, August 1968.

10. deBoor, C., A Practical Guide to Splines. Springer-Verlag, New York, 1978.
11. "IMSL Library Reference Manual," Vol. 1, Chapter I, ICSCCU and ICSEVU Routines, 9th Edition, June 1982.
12. Harder, R. L., and Desmarais, R. N., "Interpolation Using Surface Splines," Journal of Aircraft, 1972.
13. Dongarra, J., et al., LINPACK User's Guide, SIAM, Philadelphia, 1979.
14. Griffin, K. E., and Eastep, F. E., "Active Control of Forward Swept Wings with Divergence Flutter Aeroelastic Instabilities," Journal of Aircraft, Vol. 19, No. 10, October 1982, pp.885-891.
15. Nissim, E., "Flutter Suppression Using Active Controls Based on the Concept of Aerodynamic Energy," NASA TN D-6199, March 1971.
16. Nissim, E., "Active Flutter Suppression Using Trailing-Edge and Tab Control Surfaces," AIAA Journal, Vol. 14, No. 6, June 1976, pp. 757-762.
17. Rudisill, C. S., "Optimization of Active Control Systems to Suppress Flutter and Minimize Turbulence Response," AIAA Journal, Vol. 15, No. 6, June 1977, pp. 779-785.
18. Giesing, J. P., Kalman, T. P., Rodden, W. P., "Subsonic Unsteady Aerodynamics for General Configurations," AFFDL-TR-71-5, Part I, Vol. II, November 1971.

APPENDIX A
INPUT DATA INSTRUCTIONS
FOR UDEASTEX

NOTE: Input changes from previous NASA Dryden versions
are shown in **bold letters**

1.1 TITLE - 1:6 (6 lines of Title cards)

Format (FREE)

TITLE data are not included in UDFASTEX1 or UDFASTEX2 data

2.1 (IC(I), I=1,40) (Parameter inputs)

Format (10I5)

Note: Integer parameters defined as below

- IC(1) = Defines solution type
 - = 1, K type flutter analysis
 - = -1, P-K type flutter analysis
 - = 2, divergence analysis
 - = 0, pressure calculations only
- IC(2) = Defines number of vibration modes to be used in analysis
Maximum of 20
- IC(3) = Number of lifting surfaces (max. 30 for DLM, Doublet
Lattice Method, 5 for MBM, Machbox Method)
- IC(4) = Number of reduced velocities (Ki)
 - = 6 for P-K solution
 - = 30 or less for K- solution on pressure calculation
 - = 1 for divergence analysis
- IC(5) = Number of air densities (If IC(1) = 0, set IC(5) = 0)
Maximum of 10
- IC(6) = Aerodynamic forces print option
 - = 1, print data
 - = 0, no print-out is required
- IC(7) = Aerodynamic pressures print option
 - = 1, print data
 - = 0, no print-out is required
- IC(8) = Lift and moment coefficients print option
 - = 1, print data
 - = 0, no print-out is required
- IC(9) = 1, reads QBAR data (frequency independent addition to
aerodynamic matrix)
 - = 0, no addition
- IC(10) = Print option for interpolated generalized aerodynamic
forces for K- method
 - = 1, print data
 - = 0, no print-out is required

IC(11) = Index for non-zero mode numbers for normalization of flutter determinant. Any non-zero frequency is acceptable, suggest IC(11) = 1.

IC(12) = Index flutter determinant formulation
 = 1, for non-zero frequencies $\langle D = K^{-1} (M + A_E)'$
 = 0, in presence of zero frequencies $\langle D = (M + A_E)^{-1} K'$

IC(13) = Index for interpolation of aerodynamic forces
 = 1, perform interpolation
 = 0, to compute forces directly at each K_i
 (If IC(1) = -1, set IC(13) = 1)
 (If IC(1) = 1, set IC(13) = 0 or 1, as desired)
 (If IC(1) = 0 or 2, set IC(13) = 0)

IC(14) = 1, option for CALCOMP plots
 = 0, no plots

IC(15) = Index for velocity scale in flutter solution
 = 1, to use true air speed
 = 0, adopt equivalent air speed

IC(16) = Option for addition of structural damping
 = 1, indicates addition of same damping value for all modes
 = -1, indicates addition of differential damping value for each mode
 = 0, absence of added damping

IC(17) = Print option to display number of iteration for each root in P-K solution
 = 1, print data
 = 0, no print-out is required (also for IC(1)=-1)

IC(18) = Option for root extrapolation in P-K solution
 = 1, to use two previous root values
 = 0, to use previous root value (also for IC(1)=-1)

IC(19) = Index for ordering of roots after P-K solution
 = 1, to perform ordering
 = 0, no ordering required (also for IC(1)=-1)

IC(20) = Print option for root iteration in flutter solution
 = 1, print iteration
 = 0, no print-out is required

IC(21) = Index to define type of aerodynamics program
 = 1, DLM option
 = 2, MBM option
 = 3, Kernel function approach

IC (22) = Index for generation and storage of A_E matrix coefficients
 = 0, to compute and save
 = 1, assumes existence of such computed data
 =-1, indicates use of MBM

IC (23) = 1, print modal input data (STARS/solids data)
 = 0, no print-out is required

IC (24) = 1, print interpolated modal data (GRIDCHG output)
 = 0, no print-out is required

IC (25) = Number of modal elimination cycles
 Maximum of 25

IC (26) = Option for stiffness variation
 = 1, to vary stiffness
 = 0, no variation required

IC (27) = Mode number for stiffness variation
 = 0, if IC (26)=0

IC (28) = Option of eigenvector display
 = 1, print data
 = 0, no print-out is required

IC (29) = 1, display vectors
 = 0, no display

IC (30) = 1, print flutter determinant matrix for K- solution
 = 0, no display

IC (31) = Option to implement changes in generalized masses and modal frequencies
 = 1, to perform changes
 = 0, no change

IC (32) = Option to revise generalized stiffness matrix
 = 1, to input changes
 = 0, no change

IC (33) = 1, for steady state analysis
 = 0, for oscillatory analysis
 (For IC (1)=2, IC (33)=1)

IC (34) = 1, option for input of scaling factor to computed aerodynamic force
 = 0, no factors

IC (35) = Print option for MBM
 = 1, print downwash in diaphragm region
 = 0, no print-out is required
 (If IC (21)=2 or IC (22)=1, set IC (35)=0)

LC(36) = Flutter redesign option
= 1, perform redesign
= 0, no redesign

LC(37) = Print option for DLM
= 1, print geometric data associated with elements
= 0, no print-out is required

LC(38) = MATDAT output file
= 10

LC(39) = Option for future use
= 0

LC(40) = Density-altitude option
= 0 for usual input of density ratios, RHOR.
= 1 for input of altitudes (feet) in place
of RHOR input.

3.1 IN (Note: IN = 1 for STARS mode input by tape)

Format (I5)

IN = 2 for input of modal data in the input file is usable in UDFASTEX1 runs. The mode shape is ignored in the UDFASTEX2 run since that program takes input directly from the output files of the MODAL_INTERPOLATION program. Include all the following four data items when IN = 2 and use is made of UDFASTEX1.

3.1.1 NC = NUMBER OF DEGREES OF FREEDOM PER MODE. FOR UDFASTEX2 RUNS SET TO ZERO AND OMIT INPUT OF ITEM 3.1.2.

3.1.2 QZ(K,I)

Modal deformations for the I-th mode and K-th degree of freedom. Format 7E10.0 for K = 1 to NC. Repeat for each mode, I=1 to LC(2). OMIT IF NC=0.

3.1.3 NCARD

Number of data lines containing all non-zero elements of the generalized mass matrix. Format I15.

3.1.4 I, J, WW(I,J)

Row, column, generalized mass matrix entry (lb.).
Format 3(2I5,E10.0). Number of lines of data is NCARD.

3.1.5 OMG(I)

Modal frequencies (Hz.). Format 7E10.3. Number of lines of data is (LC(2)-1)/7+1.

3.2 BR, FMACH (Note: BR = Reference semi-chord length, inches,
FMACH = Freestream Mach number)

Format 2F10.0)

3.3 K(I), I=1, LC(4)

Format (7F10.0)

Description: To read reduced velocities for K-method or ASE open-loop analysis

- 3.4 NV, V1, DV
Format (I5, 2F10.0)
Note: This data is required only for P-K Solution
NV = Number of velocities for initial analysis (maximum = 20)
V1 = Lowest velocity in knots (suggest at least 200)
DV = Velocity increment (suggest less than 250)
- 3.5 TOL, <RVBO(I), I=1, 6'
Format (7F10.0)
Note: TOL = Tolerance for interpolation fit (usually = 10^{-4})
RVBO(I) = Six reference reduced velocity values for interpolation; the first and last values are to be within range of values specified in 3.3, for K-solution. For P-K method, RVBO(1).LE.1.69*12*VMIN/9BR*WMAX and RVBO(6).GE.1.69*12*WMAX/(BR*WMIN)
VMIN = V1 knots
WMAX = V1 + (NV-1)*DV knots
WMAX, WMIN = Maximum and minimum modal frequencies (rad/sec)
- 3.6 MADD, IADD, MSYM
Format (3I5)
Note: Required only of IC(31)=1
MADD = Number of generalized mass changes
IADD = Number of changes in modal frequencies
MSYM = 0, if changes in generalized mass matrix are symmetric
= 1, if such changes are not symmetric
- 3.6.1 I, J, WW(I,J)
Format (2I5, F10.0)
Description: To input modification in generalized mass matrix
Note: I, J = Row and column number of altered generalized mass
WW(I,J) = Altered mass value (If MSYM = 0, enter only upper triangular masses)
3.6.1 is repeated for K=1, MADD
- 3.6.2 I, CMG(I)
Format (I5, F10.0)
Description: Input for frequency modifications
Note: I = Mode number to be changed
CMG(I) = New frequency value (HZ)
3.6.2 is repeated for K=1, IADD
- 3.6.3 GMAX, GMIN, VMAX, FMAX
Format (4F10.0)
Note: GMAX = Minimum value of damping scale for V-g, V-f plots
GMIN = Minimum value of damping scale for V-g, V-f plots
VMAX = Maximum value of velocity scale in knots
FMAX = Maximum value of frequency scale in HZ

3.6.4 RHOR(I), I=1, LC(5)

FORMAT (7F10.0)

NOTE: RHOR(I) = denoting ratio with reference to sea level for flutter and divergence analysis, a maximum of 10

NOTE: RHOR(I) = altitudes in feet if LC(40) = 1. Conversion back to density ratios is made by call to subroutine DENSITY.

3.7 NOTIR, (NINZ(J,I), J=1,NOTIR), I=1, LC(25)

Format (10I5)

Description: To read modal elimination data set, LC(25) times

Note: NOTIR = Number of modes to be eliminated

NINZ(J,I) = Mode number of modes to be eliminated

3.8 FL, ACAP

Format (2F10.0)

Note: FL = Reference chord length in inches

ACAP = Reference area in (inch)²

3.9 NDELT, NP, NB, NCORE, N3, N4, N7

Format (7I5)

Description: Data on doublet lattice aerodynamic paneling

Note: NDELT = 1, aerodynamics are symmetrical about Y=0

= -1, aerodynamics are antisymmetrical about Y=0

= 0, no symmetry along Y=0 (single surface)

NP = total number of 'panels' on all lifting 'surfaces' and all optional interacting bodies, a maximum of 50. Each surface is divided into major trapezoidal subdivisions called panels, based upon geometrical discontinuities. The parallel edges are parallel to the free stream.

NB = number of bodies that aerodynamically interact with the surfaces (maximum = 20)

NCORE = problem size, N * M
N is the total number of aerodynamic elements (maximum = 400)
M is the number of modes

N3 = 1, display pressure influence coefficients
= 0, no display

N4 = 1, display influence coefficients relating downwash on lifting surfaces to body element pressures

N7 = 1, calculate pressures and generalized forces
= 0, cease computation after influence coefficients are determined

4.1 XOD(I), YO(I), ZO(I), GGMAS(I)
Format (4F10.0)

Description: To translate panels back to original location, perturbed originally in GRIDCHG program. Such coordinates are in the Global (Aircraft) system indicating position of origin of the LCS for each panel.

Note: GGMAS(I) = panel dihedral or V.T. rotation
= 0 for interference panel

4.1.1 X1, X2, X3, X4, Y1, H2
Format (6F10.)

Description: The coordinates (X1, Y1) and (X2, Y1) are for the leading and trailing inboard corners of the panels. The coordinates (X3, Y2) and (X4, Y2) are the leading and trailing outboard corners in LCS.

4.1.2 Z1, Z2, NS, NC, COEFF
Format (2F10.0, 1X, 2I3, 3X, 1F10.0)

Description: Defines boundaries of 'elements' in the panel. The panel is divided into a number of smaller trapezoids, called elements, by lines of constant percent panel panel chord and of constant percent panel span.

Note: Z1 = Z-coordinate of inboard edge of the panel in inches
Z2 = Z-coordinate of outboard edge of the panel in inches
NS = number of element boundaries in the spanwise direction
(maximum of 50)
= 2, for each body interference panel
NC = number of element boundaries in the chordwise direction
(maximum of 50)
COEFF = entered as zero

4.1.3 (TH(J), J=1,NC)
Format (6F10.0)

Description: Chordwise element boundaries for the panel in fraction of chord (TH(1)=0.0, TH(NC)=1.0)

4.1.4 (TAU(J), J=1,NS)
Format (6F10.0)

Description: Spanwise element boundaries for the panel in fraction of span (TAU(1)=0.0, TAU(NS)=1.0)

Note: The above data for section 4 is to be repeated NP times in the sequence as below:

1. Vertical panels or plane of symmetry (y=0)
2. Panels on other surfaces
3. Body interference panels

4.2 XBO(I), YBO(I), ZBO(I), I=1, NB
Format (3F10.0)

Description: Input of reference coordinates, vertically vibrating
bodies before laterally vibrating ones X, Y and Z
global reference coordinates of the body

4.2.1 ZSC, YSC, NF, NZ, NY, COEFF, MRK(I,1), MRK(I,2)
Format (2F10.0, 1X, 3I2, 3X, 1F10.0, 2I3)

Note: ZSC = local Z coordinate of the body axis
YSC = local Y coordinate of the body axis
NF = number of body element boundaries along its axis
NZ = 1, indices body vibrating in Z-direction
= 0, not vibrating in Z-direction
NY = 1, indicates body vibrating in Y-direction
= 0, indicates body not vibrating in Y-direction
(NZ = NY)
COEFF = 0.0
MRK(I,1) = index of the first element on the first interference
panel
MRK(I,2) = index of the last element on the last interference
panel

4.2.2 (F(J), J=1,NF)
Format (6F10.0)

Description: X-coordinates of the divisions of the body starting
with body nose and proceeding aft in inches (in local
coordinates)

4.2.3 (RAD(J), J=1, NF)
Format (6F10.0)

Description: Radii of body elements at the end points of division
in inches

4.3 NSTRIP, NPRI, JSPECS, NSV, NBV, NYAW
Format (6I5)

Description: Antisymmetric aerodynamics data calculations

Note: NSTRIP = number of chordwise strips of panel elements on all panels
= 1, for LC(8)=1, used for print outs of lift and moment coefficients for the strips, suggest NSTRIP=1
NPRI = 1, option to print pressures in QUAS or FUTSOL
= 0, no print out is required
JSPECS = 1, antisymmetrical aerodynamics about Z=0 (biplane or jet effect)
=-1, symmetrical aerodynamics about Z=0 (ground effect)
= 0, no symmetry about plane Z=0
NSV = number of strips lying on all vertical panels lying on the symmetric plane Y=0
NBV = number of elements on all vertical panels lying on the plane Y=0
NYAW = 0, if NDELT=1 (symmetric about Y=0)
= 1, if NDELT=-1 (antisymmetric about Y=0)
= 0 or 1, if NDELT=0

4.3.1 LIM(J,1), LIM(J,2), LIM(J,3) J=1, NSTRIP
Format (3I3)

Note: For NSTRIP 1, use a blank card
LIM(J,1) = index of first element on each chordwise strip
LIM(J,2) = index of last element on each chordwise strip
LIM(J,3) = 0

5.1 KSURF, NBOXS, NCS
Format (1I5, 2I5)

Description: Primary surface data associated with modal interpolation

Note: KSURF = T this surface has one or more control surfaces with forward hinge lines
= F this surface has no control surfaces
NBOXS = number of elements in this surface including control surfaces
NCS = number of control surfaces on primary surface (maximum of 5)

Note: Repeat this data item IC(3) times (number of surfaces) before going on to 5.1.1

-----` IMPORTANT CHANGE \-----

OMIT ALL THE REMAINING ITEMS EXCEPT 5.4 IF USING UDFASTEX2 VERSION.
MODE INTERPOLATION IS PERFORMED EXTERNALLY USING A SURFACE SPLINE.
CONTROL SURFACES ARE TREATED AS SEPARATE AERODYNAMIC SURFACES.

-----` IMPORTANT CHANGE \-----

5.1.1 NLines, NELAXS, NICH, NISP

Format (4I5)

Note: NLines = number of lines on this primary surface along which
nodal data are input (maximum of 20)
NELAXS = 1, translation and pitch rotation are prescribed at
each input point
= 0, only translation is prescribed
NICH = control word option for the type of extrapolation done
in the chordwise direction in interpolating modal data
in the aerodynamic grid
= 0, linear
= 1, quadratic
= 2, cubic
NISP = control word option for the type of extrapolation done
in the spanwise direction in interpolating modal data
to the aerodynamic grid
= 0, linear
= 1, quadratic
= 2, cubic

5.1.2 NGP(I), XTERM1(I), YTERM1(I), XTERM2(I), YTERM2(I)

Format (I5, 4F10.0)

Note: NGP(I) = number of points on the Ith line of primary surface
at which the nodal data are specified
(maximum number is 12)
XTERM1(I) = x-coordinate of the inboard terminus of the Ith
line for the primary surface, in inches, defined
in ICS
YTERM1(I) = y-coordinate of the inboard terminus of the Ith
line for the primary surface, in inches, defined
in ICS
XTERM2(I) = x-coordinate of the outboard terminus of the Ith
line for the primary surface, in inches, defined
in ICS
YTERM2(I) = y-coordinate of the outboard terminus of the Ith
line for the primary surface, in inches, defined
in ICS

5.1.3 (YGP(J,I), J=1,NGP(I))

Format (8F10.0)

Note: YGP(J,I) spanwise coordinates of the points along the Ith line at which input modal data are given in inches in LCS. Start with the most inboard point and proceed outboard.

Note: Repeat data input in 5.1.2 and 5.1.3 for I=1, NLINES

5.1.4 DIST

FORMAT (E10.0)

Description: An arbitrary chordwise distance for a primary surface from the given line to a reference line on which modal displacements are calculated in inches (input if NELAXS=1)

5.2 (X1(J), Y1(J), X2(J), Y2(J), J=1, NCS)

Format (4F10.0)

Description: hinge line data for control surfaces. To enter data only if KSURF=T

Note: X1(J) = x-coordinate of inboard terminus of the Jth control surface leading edge in inches in LCS

Y1(J) = y-coordinate of inboard terminus of the Jth control surface leading edge in inches in LCS

X2(J) = x-coordinate of the outboard terminus of the Jth control surface leading edge in inches in LCS

Y2(J) = y-coordinate of the outboard terminus of the Jth control surface leading edge in inches in LCS

5.2.1 NLINES, NELAXS, NICH, NISP

Format(4I5)

Note: NLINES = number of lines on this control surface along which modal data are input (maximum of 20)

NELAXS = 1, translation and pitch rotation are prescribed at each input point

= 0, only translation is prescribed

NICH = control word option for the type of extrapolation done in the chordwise direction in interpolating modal data in the aerodynamic grid

= 0, linear

= 1, quadratic

= 2, cubic

NISP = control word option for the type of extrapolation done in the spanwise direction in interpolating modal data to the aerodynamic grid

= 0, linear

= 1, quadratic

= 2, cubic

5.2.2 NGP(I), XTERM1(I), YTERM1(I), XTERM2(I), YTERM2(I)

Format (I5, 4F10.0)

Note: NGP(I) = number of points on the Ith line of control surface at which the nodal data are specified (maximum number is 12)

XTERM1(I) = x-coordinate of the inboard terminus of the Ith line for the primary surface, in inches, defined in LCS

YTERM1(I) = y-coordinate of the inboard terminus of the Ith line for the primary surface, in inches, defined in LCS

XTERM2(I) = x-coordinate of the outboard terminus of the Ith line for the primary surface, in inches, defined in LCS

YTERM2(I) = y-coordinate of the outboard terminus of the Ith line for the primary surface, in inches, defined in LCS

5.2.3 (YGP(J,I), J=1, NGP(I))

Format (8F10.0)

Note: YGP(J,I) spanwise coordinates of the points along the Ith line at which input modal data are given in inches in LCS. Start with the most inboard point and proceed outboard.

Note: Repeat data input in 5.2.2 and 5.2.3 for I=1, NLINES

5.2.4 DIST

Format (E10.0)

Description: An arbitrary chordwise distance for a control surface from the given line to a reference line on which modal displacements are calculated in inches (input if NELAXS=1)

5.3.1 INGP, NSTRIP, IPANEL

Format (3I5)

Description: Body surface data associated with modal interpolation

Note: INGP = number of points on the Jth body axis at which modal data are prescribed (maximum of 20)

NSTRIP = number of interference panels (or strips) associated with the Jth body

IPANEL = index of the first such interference panel associated with the Jth body

5.3.2 (XGP(J), J=1, NGP)

Format (6F10.0)

Note: XGP(J) = streamwise coordinates of each point at which modal data are prescribed in inches in LCS

Note: To repeat 5.3.1 and 5.3.2 for each body, i.e., NB times

5.4 KLUGLB

Format (I5)

Note: KLUGLB = 1, print global geometry
= 0, no print out is required

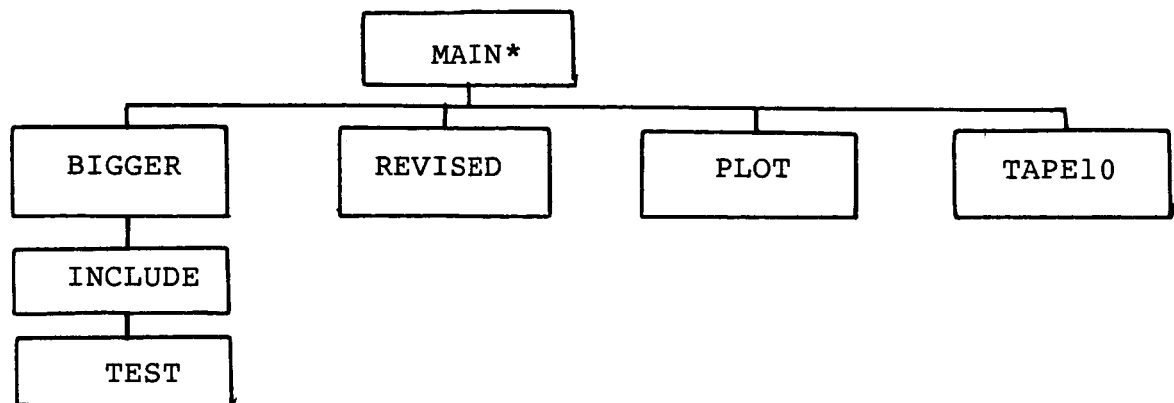
APPENDIX B

PROCEDURE FILE DOCUMENTATION

PROCEDURE FILE DOCUMENTATION

The purpose of this appendix is to document the directory and file structures of the main UD.COM procedure file discussed in Section 5. The directory structure is shown in Figure B.1. A description of the directories is given in Table B.1. Notes in this table refer to Table 5.1 which lists the UDFASTEX menu options.

A listing of the UD.COM procedure is also included in this appendix.



*[DRYDEN.RFT] directory name used on University of Dayton
VAX 11/780

Figure B.1. Directory Structure of UDFASTEX Codes.

TABLE B.1
UDFASTEX DIRECTORIES

<u>Directory</u>	<u>Description</u>
MAIN	Main directory for all programs, data, and procedure files developed. Directory [DRYDEN.RFT] name used on University of Dayton Research Institute VAX 11/780.
BIGGER	Development directory used for the dynamic dimensioning of UDFASTEX.
INCLUDE	<p>Contains all UDFASTEX Fortran codes, data, and procedure files for Options 1, 2, 7, 8, and 9 shown on Table 5.1. The Fortran codes contain "include" statements of the form</p> <p style="text-align: center;">Name.INS</p> <p>where "Name" is the subroutine or function name for the respective routines.</p>
TEST	<p>Contains all UDFASTEX Fortran codes, data, and procedures for Option 6 shown in Table 5.1. Files of the form</p> <p style="text-align: center;">Name.\$NS</p> <p>are edited and copied into the INCLUDE directory to become the new</p> <p style="text-align: center;">Name.INS</p> <p>files for the dynamically dimensioned UDFASTEX code.</p>
REVISED	Development directory used for the new modal interpolation routines used in UDFASTEX.

MODE	Contains all UDFASTEX Fortran codes, data, and procedure files for Option 3 shown on Table 5.1.
PLOT	Contains all UDFASTEX Fortran codes, data, and procedure files for Option 5 shown on Table 5.1.
TAPE10	Contains all UDFASTEX Fortran codes, data, and procedure files for Option 4 shown on Table 5.1.

LISTING OF MAIN UDFASTEX PROCEDURE FILE : UD.COM
 DIRECTORY NAMES ARE THOSE USED ON THE UNIVERSITY OF DAYTON
 RESEARCH INSTITUTE VAX 11/780

```
$ !
$ SET DEF USER_DISK:[DRYDEN.RFT]
$ !
$ ON CONTROL_Y THEN GOTO MAIN_MENU_NO_CLEAR
$ ON ERROR THEN GOTO MAIN_MENU_NO_CLEAR
$ ON WARNING THEN GOTO MAIN_MENU_NO_CLEAR
$ !
$ TT      == WRITE SYS$OUTPUT
$ CLEAR   == SET TERM / WIDTH = 80
$ GOTOMAIN == SET DEF [DRYDEN.RFT]
$ GOTOINCLUDE == SET DEF [DRYDEN.RFT.BIGGER.INCLUDE]
$ GOTOMODE  == SET DEF [DRYDEN.RFT.REVISED.MODE]
$ GOTOPLOT  == SET DEF [DRYDEN.RFT.PLOT]
$ GOTOTAPE10 == SET DEF [DRYDEN.RFT.TAPE10]
$ GOTOTEST  == SET DEF [DRYDEN.RFT.BIGGER.INCLUDE.TEST]
$ !
$ CLEAR
$ TT " "
$ TT " "
$ TT " "
$ TT " "
$ TT " "
$ TT " "
$ TT " "
$ TT " "
$ TT " "
$ TT " -----"
$ TT " -----"
$ TT " MASTER PROCEDURE ACTIVATED FOR RUNNING"
$ TT " ALL REVISED FASTEX FLUTTER ANALYSIS CODES"
$ TT " DEVELOPED BY THE UNIVERSITY OF DAYTON"
$ TT " FOR NASA DRYDEN FLIGHT RESEARCH CENTER"
$ TT " UNDER GRANT NAG 2-377"
$ TT " ( LAST REVISED FEBRUARY 6, 1987 )"
$ TT " -----"
$ TT " -----"
$ TT " "
$ INQUIRE YN -
$ " DO YOU WANT TO CONTINUE(Y/N)?
$ IF ( YN .EQ. "Y" ) THEN GOTO MAIN_MENU
$ GOTO QUIT
$ ! *****
$ MAIN_MENU:
$ ! *****
$ CLEAR
$ MAIN_MENU_NO_CLEAR:
$ TT " "
$ TT " "
```



```

$ TT " -----"
$ TT " UDFASTEX MAIN MENU"
$ TT " -----"
$ TT " OPTION DESCRIPTION"
$ TT " 1 Run UDFASTEX with internal mode interpolation"
$ TT " 2 Run UDFASTEX with external mode interpolation"
$ TT " 3 Compute aero box data or run plate spline"
$ TT " 4 Run an alternate aero or eigen solution"
$ TT " 5 Plot flutter V-g or V-f data"
$ TT " 6 Change the problem size capacity of UDFASTEX"
$ TT " 7 Compile and link an new version of UDFASTEX"
$ TT " for OPTION 1 use"
$ TT " 8 Compile and link an new version of UDFASTEX"
$ TT " for OPTION 2 use"
$ TT " 9 Create a print file of selected UDFASTEX output"
$ TT " 99 Quit"
$ TT " "
$ INQUIRE OPT -
$ " ENTER OPTION"
$ IF ( OPT .EQ. "1" ) THEN GOTO RUN_INTERNAL
$ IF ( OPT .EQ. "2" ) THEN GOTO RUN_EXTERNAL
$ IF ( OPT .EQ. "3" ) THEN GOTO AERO_SPLINE
$ IF ( OPT .EQ. "4" ) THEN GOTO AERO_EIGEN
$ IF ( OPT .EQ. "5" ) THEN GOTO PLOT
$ IF ( OPT .EQ. "6" ) THEN GOTO CHANGE_SIZE
$ IF ( OPT .EQ. "7" ) THEN GOTO COMPILE_AND_LINK1
$ IF ( OPT .EQ. "8" ) THEN GOTO COMPILE_AND_LINK2
$ IF ( OPT .EQ. "9" ) THEN GOTO SELECT_PRINT
$ IF ( OPT .EQ. "99" ) THEN GOTO QUIT
$ TT " INVALID OPTION - TRY AGAIN..."
$ GOTO MAIN_MENU:
$ ! *****
$ ! *****
$ RUN_INTERNAL:
$ ! OPTION 1 Run UDFASTEX with internal mode interpolation
$ ! *****
$ CLEAR
$ TT " "
$ TT " OPTION 1 RUN FASTEX WITH INTERNAL MODE INTERPOLATION"
$ TT " IS STARTING..."
$ TT " "
$ OPT1:
$ TT " "
$ GOTOINCLUDE
$ ON ERROR THEN GOTO MAIN_MENU_NO_CLEAR
$ @OPTION_1_REQUIREMENTS
$ TT " "
$ TT " DO YOU WANT TO"
$ TT " 1) CHECK DIRECTORY"
$ TT " 2) CONTINUE"
$ TT " 3) GO BACK TO MAIN MENU"
$ INQUIRE OPT -
$ " ENTER OPTION"
$ IF ( OPT .EQ. "1" ) THEN GOTO OPT11

```

```

$ IF ( OPT .EQ. "2" ) THEN GOTO OPT12
$ IF ( OPT .EQ. "3" ) THEN GOTO OPT13
$ ! INVALID ENTRY
$ TT "      INVALID OPTION - TRY AGAIN..."
$ GOTO OPT1
$ OPT11:      ! CHECK FOR FILES
$ DIR .DAT
$ DIR .EXE
$ GOTO OPT1
$ OPT12:      ! START PROCEDURES
$ ON ERROR      THEN GOTO  MAIN_MENU_NO_CLEAR
$ @START_OPTION_1
$ GOTOMAIN
$ GOTO OPT1
$ OPT13:      ! GO BACK TO MAIN MENU
$ GOTO MAIN_MENU
$ ! *****
$ ! *****
$
$                      RUN_EXTERNAL:
$ ! OPTION 2      Run UDFASTEX with external mode interpolation
$ ! *****
$ CLEAR
$ TT " "
$ TT " OPTION 2 RUN UDFASTEX WITH EXTERNAL MODE INTERPOLATION"
$ TT "      IS STARTING..."
$ TT " "
$ OPT2:
$ TT " "
$ GOTOINCLUDE
$ ON ERROR      THEN GOTO  MAIN_MENU_NO_CLEAR
$ @OPTION_2_REQUIREMENTS
$ TT " "
$ TT " DO YOU WANT TO"
$ TT "      1) CHECK DIRECTORY"
$ TT "      2) CONTINUE"
$ TT "      3) GO BACK TO MAIN MENU"
$ INQUIRE OPT -
$      " ENTER OPTION"
$ IF ( OPT .EQ. "1" ) THEN GOTO OPT21
$ IF ( OPT .EQ. "2" ) THEN GOTO OPT22
$ IF ( OPT .EQ. "3" ) THEN GOTO OPT23
$ ! INVALID ENTRY
$ TT "      INVALID OPTION - TRY AGAIN..."
$ GOTO OPT2
$ OPT21:      ! CHECK FOR FILES
$ DIR .DAT
$ DIR .EXE
$ GOTO OPT2
$ OPT22:      ! START PROCEDURES
$ ON ERROR      THEN GOTO  MAIN_MENU_NO_CLEAR
$ @START_OPTION_2
$ GOTOMAIN
$ GOTO OPT2
$ OPT23:      ! GO BACK TO MAIN MENU

```

```

$ GOTO MAIN_MENU
$ ! *****
$ ! *****
$
$           AERO_SPLINE:
$ ! OPTION 3      Compute aero box data or run plate spline
$ ! *****
$ CLEAR
$ TT " "
$ TT " OPTION 3 COMPUTE AERO BOX DATA OR RUN PLATE SPLINE"
$ TT "           IS STARTING..."
$ TT " "
$ OPT3:
$ TT " "
$ GOTOMODE
$ ON ERROR      THEN GOTO  MAIN_MENU_NO_CLEAR
$ @OPTION_3_REQUIREMENTS
$ TT " "
$ TT " DO YOU WANT TO"
$ TT "   1) CHECK DIRECTORY"
$ TT "   2) CONTINUE"
$ TT "   3) GO BACK TO MAIN MENU"
$ INQUIRE OPT -
$   " ENTER OPTION"
$ IF ( OPT .EQ. "1" ) THEN GOTO OPT31
$ IF ( OPT .EQ. "2" ) THEN GOTO OPT32
$ IF ( OPT .EQ. "3" ) THEN GOTO OPT33
$ ! INVALID ENTRY
$ TT "   INVALID OPTION - TRY AGAIN..."
$ GOTO OPT3
$ OPT31:      ! CHECK FOR FILES
$ DIR .DAT
$ DIR .EXE
$ GOTO OPT3
$ OPT32:      ! START PROCEDURES
$ ON ERROR      THEN GOTO  MAIN_MENU_NO_CLEAR
$ @START_OPTION_3
$ GOTOMAIN
$ GOTO OPT3
$ OPT33:      ! GO BACK TO MAIN MENU
$ GOTO MAIN_MENU
$ ! *****
$ ! *****
$
$           AERO_EIGEN:
$ ! OPTION 4      Run an alternate aero or eigen solution
$ ! *****
$ CLEAR
$ TT " "
$ TT " OPTION 4 RUN AN ALTERNATE AERO OR EIGEN SOLUTION"
$ TT "           IS STARTING..."
$ TT " "
$ OPT4:
$ TT " "
$ GOTOTAPE10
$ ON ERROR      THEN GOTO  MAIN_MENU_NO_CLEAR

```

```

$ @OPTION_4_REQUIREMENTS
$ TT " "
$ TT " DO YOU WANT TO"
$ TT " 1) CHECK DIRECTORY"
$ TT " 2) CONTINUE"
$ TT " 3) GO BACK TO MAIN MENU"
$ INQUIRE OPT -
    " ENTER OPTION"
$ IF ( OPT .EQ. "1" ) THEN GOTO OPT41
$ IF ( OPT .EQ. "2" ) THEN GOTO OPT42
$ IF ( OPT .EQ. "3" ) THEN GOTO OPT43
$ ! INVALID ENTRY
$ TT "      INVALID OPTION - TRY AGAIN..."
$ GOTO OPT4
$ OPT41:      ! CHECK FOR FILES
$ DIR .DAT
$ DIR .EXE
$ GOTO OPT4
$ OPT42:      ! START PROCEDURES
$ ON ERROR      THEN GOTO  MAIN_MENU_NO_CLEAR
$ @START_OPTION_4
$ GOTOMAIN
$ GOTO OPT4
$ OPT43:      ! GO BACK TO MAIN MENU
$ GOTO MAIN_MENU
$ ! *****
$ ! *****
$
$          PLOT:
$ ! OPTION 5      Plot flutter V-g or V-f data
$ ! *****
$ CLEAR
$ TT " "
$ TT " OPTION 5 PLOT FLUTTER V-G OR V-F DATA"
$ TT "      IS STARTING..."
$ TT " "
$ OPT5:
$ TT " "
$ GOTO PLOT
$ ON ERROR      THEN GOTO  MAIN_MENU_NO_CLEAR
$ @OPTION_5_REQUIREMENTS
$ TT " "
$ TT " DO YOU WANT TO"
$ TT " 1) CHECK DIRECTORY"
$ TT " 2) CONTINUE"
$ TT " 3) GO BACK TO MAIN MENU"
$ INQUIRE OPT -
    " ENTER OPTION"
$ IF ( OPT .EQ. "1" ) THEN GOTO OPT51
$ IF ( OPT .EQ. "2" ) THEN GOTO OPT52
$ IF ( OPT .EQ. "3" ) THEN GOTO OPT53
$ ! INVALID ENTRY
$ TT "      INVALID OPTION - TRY AGAIN..."
$ GOTO OPT5
$ OPT51:      ! CHECK FOR FILES

```

```

$ DIR FLUTTERPLOTFILE.DAT
$ GOTO OPT5
$ OPT52:      ! START PROCEDURES
$ ON ERROR    THEN GOTO  MAIN_MENU_NO_CLEAR
$ @START_OPTION_5
$ GOTOMAIN
$ GOTO OPT5
$ OPT53:      ! GO BACK TO MAIN MENU
$ GOTO MAIN_MENU
$ ! *****
$ ! *****
$          CHANGE_SIZE:
$ ! OPTION 6   Change the problem_size capacity of UDFASTEX
$ ! *****
$ CLEAR
$ TT " "
$ TT " OPTION 6 CHANGE THE PROBLEM SIZE CAPACITY OF UDFASTEX"
$ TT "          IS STARTING..."
$ TT " "
$ OPT6:
$ TT " "
$ GOTOTEST
$ ON ERROR    THEN GOTO  MAIN_MENU_NO_CLEAR
$ @OPTION_6_REQUIREMENTS
$ TT " "
$ TT " DO YOU WANT TO"
$ TT "   1) CHECK DIRECTORY"
$ TT "   2) CONTINUE"
$ TT "   3) GO BACK TO MAIN MENU"
$ INQUIRE OPT -
$   " ENTER OPTION"
$ IF ( OPT .EQ. "1" ) THEN GOTO OPT61
$ IF ( OPT .EQ. "2" ) THEN GOTO OPT62
$ IF ( OPT .EQ. "3" ) THEN GOTO OPT63
$ ! INVALID ENTRY
$ TT "      INVALID OPTION - TRY AGAIN..."
$ GOTO OPT6
$ OPT61:      ! CHECK FOR FILES
$ DIR .$NS
$ GOTO OPT6
$ OPT62:      ! START PROCEDURES
$ ON ERROR    THEN GOTO  MAIN_MENU_NO_CLEAR
$ @START_OPTION_6
$ GOTOMAIN
$ GOTO OPT6
$ OPT63:      ! GO BACK TO MAIN MENU
$ GOTO MAIN_MENU
$ ! *****
$ ! *****
$          COMPILE_AND_LINK1:
$ ! OPTION 7   Compile and link an new version of UDFASTEX
$ !           for OPTION 1 use
$ ! *****
$ CLEAR

```

```

$ TT " "
$ TT " OPTION 7 COMPILE AND LINK A NEW VERSION OF UDFASTEX1"
$ TT "      IS STARTING..."
$ TT " "
$ OPT7:
$ TT " "
$ GOTOINCLUDE
$ ON ERROR      THEN GOTO  MAIN_MENU_NO_CLEAR
$ @OPTION_7_REQUIREMENTS
$ TT " "
$ TT " DO YOU WANT TO"
$ TT "      1) CHECK DIRECTORY"
$ TT "      2) CONTINUE"
$ TT "      3) GO BACK TO MAIN MENU"
$ INQUIRE OPT -
      " ENTER OPTION"
$ IF ( OPT .EQ. "1" ) THEN GOTO OPT71
$ IF ( OPT .EQ. "2" ) THEN GOTO OPT72
$ IF ( OPT .EQ. "3" ) THEN GOTO OPT73
$ ! INVALID ENTRY
$ TT "      INVALID OPTION - TRY AGAIN..."
$ GOTO OPT7
$ OPT71:      ! CHECK FOR FILES
$ DIR .FOR
$ DIR .INS
$ GOTO OPT7
$ OPT72:      ! START PROCEDURES
$ ON ERROR      THEN GOTO  MAIN_MENU_NO_CLEAR
$ @START_OPTION_7
$ GOTOMAIN
$ GOTO OPT7
$ OPT73:      ! GO BACK TO MAIN MENU
$ GOTO MAIN_MENU
$ ! *****
$ ! *****
$      COMPILE_AND_LINK2:
$ ! OPTION 8      Compile and link an new version of UDFASTEX
$ !      for OPTION 2 use
$ ! *****
$ CLEAR
$ TT " "
$ TT " OPTION 8 COMPILE AND LINK A NEW VERSION OF UDFASTEX2"
$ TT "      IS STARTING..."
$ TT " "
$ OPT8:
$ TT " "
$ GOTOINCLUDE
$ ON ERROR      THEN GOTO  MAIN_MENU_NO_CLEAR
$ @OPTION_8_REQUIREMENTS
$ TT " "
$ TT " DO YOU WANT TO"
$ TT "      1) CHECK DIRECTORY"
$ TT "      2) CONTINUE"
$ TT "      3) GO BACK TO MAIN MENU"

```

```

$ INQUIRE OPT -
    " ENTER OPTION"
$ IF ( OPT .EQ. "1" ) THEN GOTO OPT81
$ IF ( OPT .EQ. "2" ) THEN GOTO OPT82
$ IF ( OPT .EQ. "3" ) THEN GOTO OPT83
$ ! INVALID ENTRY
$ TT "      INVALID OPTION - TRY AGAIN..."
$ GOTO OPT8
$ OPT81:      ! CHECK FOR FILES
$ DIR .FOR
$ DIR .INS
$ GOTO OPT8
$ OPT82:      ! START PROCEDURES
$ ON ERROR      THEN GOTO  MAIN_MENU_NO_CLEAR
$ @START_OPTION_8
$ GOTOMAIN
$ GOTO OPT8
$ OPT83:      ! GO BACK TO MAIN MENU
$ GOTO MAIN_MENU
$ ! *****
$ ! *****
$          SELECT_PRINT:
$ ! OPTION 9      Create a print file of selected UDFASTEX output"
$ ! *****
$ CLEAR
$ TT " "
$ TT " OPTION 9 CREATE A PRINT FILE OF SELECTED UDFASTEX OUTPUT"
$ TT "      IS STARTING..."
$ TT " "
$ OPT9:
$ TT " "
$ GOTOINCLUDE
$ ON ERROR      THEN GOTO  MAIN_MENU_NO_CLEAR
$ @OPTION_9_REQUIREMENTS
$ TT " "
$ TT " DO YOU WANT TO"
$ TT "      1) CHECK DIRECTORY"
$ TT "      2) CONTINUE"
$ TT "      3) GO BACK TO MAIN MENU"
$ INQUIRE OPT -
    " ENTER OPTION"
$ IF ( OPT .EQ. "1" ) THEN GOTO OPT91
$ IF ( OPT .EQ. "2" ) THEN GOTO OPT92
$ IF ( OPT .EQ. "3" ) THEN GOTO OPT93
$ ! INVALID ENTRY
$ TT "      INVALID OPTION - TRY AGAIN..."
$ GOTO OPT9
$ OPT91:      ! CHECK FOR FILES
$ DIR UDFAS*.LIS
$ GOTO OPT9
$ OPT92:      ! START PROCEDURES
$ ON ERROR      THEN GOTO  MAIN_MENU_NO_CLEAR
$ @START_OPTION_9
$ GOTOMAIN

```

```
$ GOTO OPT9
$ OPT93:      ! GO BACK TO MAIN MENU
$ GOTO MAIN_MENU
$ ! *****
$ ! *****
$
$              QUIT:
$ ! OPTION 99      Quit
$ ! *****
$ GOTOMAIN
$ EXIT
```


APPENDIX C
PLATE SPLINE THEORY

BIVARIATE DATA INTERPOLATION

The following is based on the theory presented in Reference 12.

1. Usage -

- a. Calculation of modal data in aerodynamics grid.
- b. Calculation of aerodynamic matrices at intermediate values of reduced frequency and Mach number.

2. Problem Statement -

Given

$$(x_j, y_j, W_j) \quad j=1, \dots, N$$

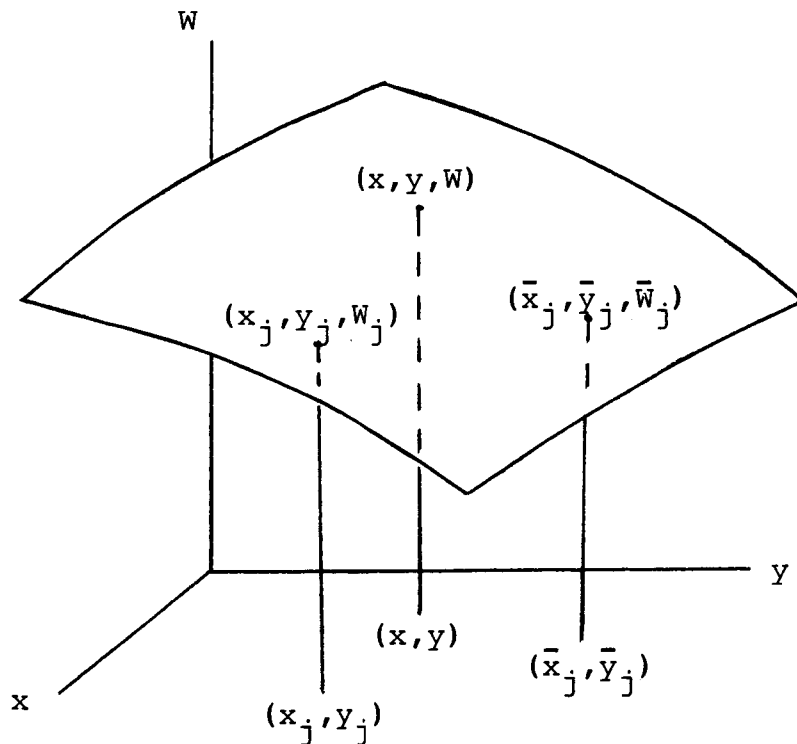
$$(\bar{x}_j, \bar{y}_j) \quad j=1, \dots, M$$

fit a mathematical function, $W=W(x,y)$ through the W_j points and compute

$$W(\bar{x}_j, \bar{y}_j) \equiv \bar{W}_j$$

$$\frac{\partial}{\partial x} W(\bar{x}_j, \bar{y}_j) \equiv \bar{W}_{xj} \quad j=1, \dots, M$$

$$\frac{\partial}{\partial y} W(\bar{x}_j, \bar{y}_j) \equiv \bar{W}_{yj}$$



3. Equations -

Assume the plate spline function in the form

$$(1) \quad W(x,y) = a_0 + a_1x + a_2y + \sum_{i=1}^N K_i(x,y)P_i$$

where

$$(2) \quad K_i(x,y) = r_i^2 \ln r_i^2$$

$$(3) \quad r_i^2 = (x - x_i)^2 + (y - y_i)^2$$

and $a_0, a_1, a_2, P_1, \dots, P_N$ are constants to be determined by the conditions

$$(4) \quad \sum_{i=1}^N P_i = 0$$

$$(5) \quad \sum_{i=1}^N x_i P_i = 0$$

$$(6) \quad \sum_{i=1}^N y_i P_i = 0$$

$$(7) \quad W_j = a_0 + a_1x_j + a_2y_j + \sum_{i=1}^N K_{ij}P_i, \quad j=1, \dots, N$$

where

$$(8) \quad K_{ij} = \begin{cases} 0 & i=j \\ K_i(x_j, y_j) & i \neq j \end{cases}$$

Note $K_{ij} = K_{ji}$.

Write (4) - (7) in the matrix form of $N+3$ equations in $N+3$ unknowns:

$$(9) \quad [A]\{c\} = \{b\}$$

where the spline matrix is

$$(10) \quad [A] = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & \dots & 1 \\ 0 & 0 & 0 & x_1 & x_2 & \dots & x_N \\ 0 & 0 & 0 & y_1 & y_2 & \dots & y_N \\ \hline 1 & x_1 & y_1 & 0 & K_{12} & \dots & K_{1N} \\ 1 & x_2 & y_2 & K_{21} & 0 & \dots & K_{2N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & y_N & K_{N1} & K_{N2} & \dots & 0 \end{bmatrix}$$

N+3 by N+3

$$(11) \quad \{b\} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ -\frac{1}{W_1} \\ W_2 \\ \vdots \\ W_N \end{Bmatrix}$$

N+3 by 1

and the desired solution is the set of spline coefficients

$$(12) \quad \{c\} = \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \\ p_1 \\ p_2 \\ \vdots \\ p_N \end{Bmatrix}$$

N+3 by 1

The required derivatives of W are

$$(13) \quad \frac{\partial W}{\partial x} = a_1 + \sum_{i=1}^N K_{xi}(x,y) P_i$$

where

$$(14) \quad K_{xi}(x,y) = 2(x - x_i)(1 + \ln r_i^2)$$

and

$$(15) \quad \frac{\partial W}{\partial y} = a_2 + \sum_{i=1}^N K_{yi}(x,y) P_i$$

where

$$(16) \quad K_{yi}(x,y) = 2(y - y_i)(1 + \ln r_i^2)$$

4. Computational Procedure

<u>STEP</u>	<u>COMMENTS</u>
1. Input N x_1, \dots, x_N y_1, \dots, y_N w_1, \dots, w_N	N = Number of given data points x_1, \dots, x_N = x coordinates of given data y_1, \dots, y_N = y coordinates of given data w_1, \dots, w_N = Given functional values
2. Input M $\bar{x}_1, \dots, \bar{x}_M$ $\bar{y}_1, \dots, \bar{y}_M$	M = Number of points at which W, w_x , w_y are needed $\bar{x}_1, \dots, \bar{x}_M$ = x coordinates of needed points $\bar{y}_1, \dots, \bar{y}_M$ = y coordinates of needed points
3. Compute K_{ij} by Equation (8)	K_{ij} = Spline weights. Compute $K_{ij} = 0$ separately, \ln terms become large otherwise.

<u>STEP</u>	<u>COMMENTS</u>
4. Set up [A] matrix by Equation (10)	Note [A] is symmetric
5. Solve Equations (3) for {c}	If Gauss elimination is used, note that diagonals are zero. Use symmetry of [A] in solution process if possible.
6. Use {c} in Equation (1) to compute \bar{w}_j	\bar{w}_j = desired functional values
7. Use {c} in Equation (13) to get \bar{w}_{xj}	\bar{w}_{xj} = x derivatives desired
8. Use {c} in Equation (15) to get \bar{w}_{yj}	\bar{w}_{yj} = y derivatives desired

5. Extensions -

Improved accuracy may be obtained by coordinate transformations on the (x_j, y_j) , (\bar{x}_j, \bar{y}_j) grids. The following steps are proposed:

1. Translation of data to centroid

$$(17) \quad x_j' = x_j - x_0$$

$$(18) \quad y_j' = y_j - y_0$$

where

$$(19) \quad x_0 = \frac{1}{N} \sum_{j=1}^N x_j$$

$$(20) \quad y_0 = \frac{1}{N} \sum_{j=1}^N y_j$$

2. Rotation of data to principal axes

$$(21) \quad x_j'' = x_j' \cos\theta + y_j' \sin\theta$$

$$(22) \quad y_j'' = x_j' \sin\theta + y_j' \cos\theta$$

where

$$(23) \quad \theta = 1/2 \tan^{-1} \left(\frac{-2I_{x'y'}}{I_{x'} - I_{y'}} \right)$$

(Note-use ATAN2($-2I_{x'y'}$, $(I_{x'} - I_{y'})$))

$$(24) \quad I_{x'} = \sum_{j=1}^N (y_j')^2$$

$$(25) \quad I_{Y'} = \sum_{j=1}^N (x_j')^2$$

$$(26) \quad I_{x'y'} = \sum_{j=1}^N (x_j') (y_j')$$

3. Scaling of data

$$(27) \quad x_j''' = x_j''/x_{\max}''$$

$$(28) \quad y_j''' = y_j''/y_{\max}''$$

where

$$(29) \quad x_{\max}'' = \max_{j=1, N} |x_j''|$$

$$(30) \quad y_{\max}'' = \max_{j=1, N} |y_j''|$$

Apply Equations (17) - (30) to (x_j, y_j) , (\bar{x}_j, \bar{y}_j) then go to Step 3 of computational procedure.

i.e., use (x_j''', y_j''') in place of (x_j, y_j)

$(\bar{x}_j''', \bar{y}_j''')$ in place of (\bar{x}_j, \bar{y}_j)